



University of **HUDDERSFIELD**

University of Huddersfield Repository

Radhi, Hazim Esmaeel

Robust multi-criteria optimisation of welded joints

Original Citation

Radhi, Hazim Esmaeel (2014) Robust multi-criteria optimisation of welded joints. Doctoral thesis, University of Huddersfield.

This version is available at <http://eprints.hud.ac.uk/23726/>

The University Repository is a digital collection of the research output of the University, available on Open Access. Copyright and Moral Rights for the items on this site are retained by the individual author and/or other copyright owners. Users may access full items free of charge; copies of full text items generally can be reproduced, displayed or performed and given to third parties in any format or medium for personal research or study, educational or not-for-profit purposes without prior permission or charge, provided:

- The authors, title and full bibliographic details is credited in any copy;
- A hyperlink and/or URL is included for the original metadata page; and
- The content is not changed in any way.

For more information, including our policy and submission procedure, please contact the Repository Team at: E.mailbox@hud.ac.uk.

<http://eprints.hud.ac.uk/>

Robust Multi-criteria Optimisation of Welded Joints

HAZIM ESMAEEL RADHI

A Thesis submitted to University of Huddersfield in partial fulfillment of the
requirements for the degree of Doctor of Philosophy

University of Huddersfield

December 2014

ABSTRACT

Civilisation has depended on welded structures to facilitate production and improve the quality of life. Welds are used to create infrastructure upon which we rely, such as transportation, oil and gas piping, shipbuilding, bridges and buildings, and to produce the equipment that makes all of this happen. In short, the joining of two metals through welding has immensely contributed to our society.

A critical factor in the strength of welded joints is the geometry of the joints, and for this reason a robust optimisation of geometrical parameters of welded joints has been conducted in order to establish the optimum and most robust design in the presence of variation amongst geometrical parameters.

A parametric finite element analysis, using Python script, has been performed with the objective to investigate the effect of the welded joint parameters on the stress concentration factors under tensile and bending load. The results indicate that the parametric model, which is generated by Python script, can be used in a wide range of welded geometry, and has the capacity to reduce the time of computation. Additionally, an experimental study, including the geometrical identification of the welded joints, tensile test, hardness test and fatigue, has also been performed.

In order to select the best optimisation algorithms, different optimisation algorithms and performance metrics with various types of problem were examined in this study. The results from this part show the accuracy of Circumscription Metric (CM) in comparison to Pair wise Metric (PW) - which is used widely in optimisation studies. Furthermore, the results show that the Fast Multi-objective Optimisation Algorithm (FMOGA-II) outperformed other optimisation algorithms used during this study.

In this study, a new methodology for selecting the most robust designs from within the Pareto set has been developed. Finally, a traditional and robust optimisation of a butt welded joint has been performed by establishing a link between an optimisation software package and parametric finite element, the results of which show the ability of this approach to extract the robust optimal designs from the Pareto front.

DEDICATION

To the spirit of my father and mother who have given me all the support I need to be what I wanted to be and who have been sources of inspiration to me throughout my life;

To my brother and sisters for their encouragement and their support;

To my wife for her personal support, encouragement and great patience during the research period;

Finally, to my beloved children, Ali, Mayar and Hussien, who are the lights of my life;

I dedicate this work.

ACKNOWLEDGEMENT

First, I would like to express my sincere gratitude to my supervisor, Dr Simon Barrans, for his instruction, encouragement, advice and support throughout my graduate education. Thanks for trusting me and my capabilities.

I would like to acknowledge the financial support given by the Iraqi Ministry of Higher Education and Scientific Research. The efforts made by the Iraqi embassy/cultural attaché to assist with the financial and administration issues of my scholarship are really appreciated.

I am very thankful to Mrs Gwen Wood and Mr Chris Sentance and Mrs Liz Rees at the school research office.

Many thanks to my friends and colleagues at Huddersfield University who supported me in all respects during my PhD research

Finally, my biggest appreciation goes to my family, who have always been the greatest motivation for me to work, study and to live my life.

Table of Contents

ABSTRACT	2
DEDICATION	3
ACKNOWLEDGEMENT	4
TABLE OF CONTENTS	5
LIST OF FIGURES	9
LIST OF TABLES	13
LIST OF ABBREVIATIONS AND SYMBOLS	15
CHAPTER ONE: INTRODUCTION	18
1.1 Motivation for the Research	18
1.2 Aim of Study	19
1.3 Objective	20
1.4 Contribution to Knowledge	20
1.5 Structure of the Thesis	20
CHAPTER TWO: WELDING AND FATIGUE	23
2.1 Historical Background on Welding	23
2.2 Welded Joint Parameters	25
2.2.1 Geometrical parameters of welded joint	25
2.2.2 Weld imperfections	26
2.3 Optimisation of Welded Joints	27
2.3.1 Effects of welding parameters on metal properties	27
2.3.2 Effects of geometrical parameters on welded joints	31
2.4 Introduction to Fatigue	31
2.5 Fatigue Failure Stage	32
2.6 Fatigue Crack Growth Rate	34
2.7 Fatigue Loading	36
2.8 Fatigue of Welded Joints	38
2.9 Conclusion	41
CHAPTER THREE: OPTIMISATION	43
3.1 Introduction	43
3.2 Historical Development	43
3.3 Optimisation Engineering Applications	45
3.4 Optimisation Problem	46

3.4.1 Design variables	46
3.4.2 Design constraints	47
3.4.3 Objective function	49
3.5 Population-based Optimisation Methods.....	50
3.5.1 Simulated annealing optimisation.....	50
3.5.2 Particle swarm optimisation	51
3.5.3 Genetic Algorithm.....	52
3.6 Multi-objective Optimisation	55
3.6.1 Problem definition	55
3.6.2 Handling Multi-objectives.....	55
3.6.3 Pareto dominance	56
3.6.4 Classical methods	57
3.6.5 Evolutionary multi-objective optimisation	58
3.7 Robust Optimisation.....	60
3.7.1 Variability in reality.....	60
3.7.2 Robust design	61
3.7.3 Robust multi-criteria optimisation.....	62
3.8 Conclusions.....	64
CHAPTER FOUR: OPTIMISATION STUDY	66
4.1 Introduction.....	66
4.2 modeFRONTIER Software	67
4.3 Initial Population Generation.....	69
4.3.1 Design of Experiment (DoE) methods	69
4.3.2 Measuring discrepancy.....	70
4.3.3 Sobol Design	70
4.4 Measurement of Algorithm Performance	71
4.4.1 Hit-Rate Metric (HR %)	71
4.4.2 Pair Wise Metric (PW)	72
4.4.3 Circumscription Metric (CM)	72
4.4.4 Performance of distribution metrics	74
4.5 Lifting Arm Case Study	77
4.5.1 Optimisation problem formulation	77
4.5.2 Single generation optimisation	81

4.5.3 <i>Optimisation with Genetic Algorithms</i>	86
4.6 Convex and Concave Case Studies	94
4.6.1 <i>SCH problem</i>	94
4.6.2 <i>FON problem</i>	94
4.6.3 <i>Results and discussion</i>	95
4.7 Conclusion	99
CHAPTER FIVE: ROBUST OPTIMISATION	101
5.1 Introduction	101
5.2 Case Study 1: Tank Design	101
5.2.1 <i>Sensitivity analysis</i>	103
5.2.2 <i>Optimisation problem parameters</i>	104
5.3 Robust Optimisation of Lifting Arm Problem	111
5.3.1 <i>Approach 1</i>	111
5.3.2 <i>Approach 2</i>	114
5.3.3 <i>Approach 3</i>	115
5.4 Conclusion	125
CHAPTER SIX: FINITE ELEMENT METHOD	127
6.1 Fundamentals of the Finite Element Method	127
6.1.1 <i>Procedure of FEA</i>	127
6.1.2 <i>Abaqus Basics</i>	128
6.2 Modelling the Welded Joint	129
6.2.1 <i>Welded joint parameters</i>	129
6.2.2 <i>Material properties</i>	131
6.2.3 <i>Element Type</i>	131
6.2.4 <i>Load and Boundary Condition</i>	132
6.2.5 <i>Model development</i>	133
6.3 Python Model Development	136
6.3.1 <i>Python Programming language</i>	136
6.3.2 <i>ABAQUS Scripting Interface</i>	136
6.3.3 <i>Parametric model definition</i>	138
6.3.4 <i>Parametric model test</i>	140
6.4 Parametric Model Results	142
6.5 Conclusion	149

CHAPTER SEVEN: EXPERIMENTAL WORK	151
7.1 Test Specimen Preparation	152
7.2 Monotonic Tensile Testing	153
7.3 Hardness Test	156
7.4 Estimating the Geometrical Parameters of Welded Joints.....	158
7.4.1 Estimation of welded Joint geometrical parameters by using dental moulding method.....	158
7.5 Fatigue Test	162
7.5.1 Test specimen	162
7.5.2 Fatigue testing.....	162
7.6 Conclusion	166
CHAPTER EIGHT: ROBUST OPTIMIZATION OF WELDED JOINTS	168
8.1 Introduction.....	168
8.2 Integrating between modeFRONTIER and ABAQUS	168
8.2.1 Multi-objective Optimization of Two-bar Truss Problem	168
8.2.2 Multi-objective Optimization of Plate with Hole	178
8.3 Traditional Optimization of Welded Joints	183
8.3.1 Problem Formulation	183
8.3.2 Description of the Workflow	184
8.3.3 Performing Optimization.....	186
8.4 Robust Optimization of Welded Joints	189
8.4.1 The Problem Definition	190
8.4.2 Performing Optimization.....	192
8.5 Conclusion	196
CHAPTER NINE: CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK	199
9.1 Introduction.....	199
9.1.1 Main conclusions	199
9.2 Recommendations for further work	200
REFERENCES	202
APPENDICES	209
APPENDIX A	209
APPENDIX B	228
APPENDIX C	242

List of Figures

Figure 1-1: M V Kurdistan tank failure (Grawood, 1997).....	18
Figure 1-2: Disaster of the Alexander Kielland platform (Kristoffer, 2012).....	19
Figure 1-3: Hydrophone holder welded into tubular bracing of the Alexander Kielland platform (Lotsberg, 2004)	19
Figure 2-1: Geometry of a butt weld (American Welding Society, AWS A3.0M/A 3.0:2010 and British Standard Institution BS EN ISO 9692-1:2013).....	25
Figure 2-2: Defects of a butt welded joint	27
Figure 2-3: Fatigue stages on the fracture surface	34
Figure 2-4: Typical relationship between the crack growth rate and the range of stress intensity factor.	35
Figure 2-5: Nomenclature used in constant-amplitude loading (Wei & Pentti, 2003).....	36
Figure 2-6: Comparison of R-ratios for different loading (Barsom & Rolfe, 1999).....	37
Figure 2-7: Variable-amplitude loading (Wei & Pentti, 2003).....	37
Figure 2-8: Local approaches for fatigue assessment of welded joints.....	38
Figure 2-9: Reinforcement angle influence on fatigue strength of butt welds (Chapetti, 2011) ...	40
Figure 3-1: Constraint surfaces.....	49
Figure 3-2: Genetic algorithm flow chart.....	53
Figure 3-3: Main concept of Pareto dominance in a two objective problem.....	57
Figure 3-4: Block diagram of the Product/Process (Bagchi, 1993).....	61
Figure 3-5: A robust solution and optimal solution.....	62
Figure 4-1: General modeFRONTIER process of integration and optimisation modeFRONTIER document (2008)	68
Figure 4-2: Normalisation and mapping of design variables	69
Figure 4-3: Distribution of the scatter plot of (a) pseudo-random sequence and (b) pseudo-random Sobol sequence in modeFRONTIER.	71
Figure 4-4: Comparison of traditional diversity indicators	73
Figure 4-5: Case 1: regular distributions of points	75
Figure 4-6: Case 2: regular distribution of points with changing the position of red points	75
Figure 4-7: Case 3: cluster distribution of points	76
Figure 4-8: Case 4: moderate clustering of points in two dimensions.	76
Figure 4-9: Case 5: heavy clustering of points in two dimensions.....	77
Figure 4-10: Schematic diagram of the lifting arm (Ghurbal, 2003).	79
Figure 4-11: Variable and objective space for trial points generated by the Sobol sequence.	85
Figure 4-12: Definition of input and output design parameters in lifting arm problem.	86
Figure 4-13: Lifting arm project within modeFRONTIER.....	87

Figure 4-14: Comparison of the Hit-Rate value for 64 sampling points with different generations.	90
Figure 4-15: Comparison of the 5 generations results to the lifting arm problem by MOGA-II, ARMOGA, FMOGA-II, NSGA-II, MOPSO and MOSA.	91
Figure 4-16: Comparison of the 10 generations results to the lifting arm problem by MOGA-II, ARMOGA, FMOGA-II, NSGA-II, MOPSO and MOSA.	92
Figure 4-17: Comparison of the 15 generations results to the lifting arm problem by MOGA-II, ARMOGA, FMOGA-II, NSGA-II, MOPSO and MOSA.	93
Figure 4-18: The evaluated front from MOGA-II, ARMOGA, NSGA-II, FMOGA, MOSA and MOPSO for SCH problem.	97
Figure 4-19: The evaluated front from MOGA-II, ARMOGA, NSGA-II, FMOGA, MOSA and MOPSO for SCH problem.	98
Figure 5-1: Solution for the tank problem.	103
Figure 5-2: Robust optimisation of the tank problem in modeFRONTIER software	105
Figure 5-3: Spatial distribution of design variables in design space	106
Figure 5-4: History chart of objective function with 20% tolerance	107
Figure 5-5: History chart of objective function with 10% tolerance	107
Figure 5-6: History chart of objective function with 5% tolerance	108
Figure 5-7: History chart of objective function with 2.5% tolerance	108
Figure 5-8: Distribution of robust design table of FMOGA –II 100-10 with 20% tolerance	109
Figure 5-9: Distribution of robust design table of FMOGA –II 100-10 with 10% tolerance	110
Figure 5-10: Distribution of robust design table of FMOGA –II 100-10 with 5% tolerance	110
Figure 5-11: Distribution of robust design table of FMOGA –II 100-10 with 2.5% tolerance ...	111
Figure 5-12: Representation of Pareto front approximation	113
Figure 5-13: Pareto front with (+5%,-5%) constraint line	113
Figure 5-14: Pareto front with (+20%,-20%) constraint line	114
Figure 5-15: Robust optimisation of lifting arm problem in modeFRONTIER software	116
Figure 5-16: Distribution of Pareto robust design in design space for cases (1, 2 and 3)	119
Figure 5-17: Relation between standard deviation of (objective1, objective2) for cases (1, 2 and 3)	119
Figure 5-18: Distribution of Pareto robust design in design space for case (4)	120
Figure 5-19: Relation between standard deviation of (objective1, objective2) for case (4)	120
Figure 5-20: Distribution of Pareto robust design in design space for case (5)	121
Figure 5-21: Relation between standard deviation of (objective1, objective2) for case (5)	121
Figure 5-22: Distribution of Pareto robust design in design space for case (6)	122
Figure 5-23: Relation between standard deviation of (objective1, objective2) for case (6)	122

Figure 5-24: Distribution of Pareto robust design in design space for case (7).....	123
Figure 5-25: Relation between standard deviation of (objective1, objective2) for case (7)	123
Figure 5-26: Distribution of Pareto robust design in design space for case (8).....	124
Figure 5-27: Relation between standard deviation of (objective1, objective2) for case (8)	124
Figure 6-1: Abaqus stages of analysis Abaqus (2008)	129
Figure 6-2: Geometry of butt welded joint	130
Figure 6-3: Abaqus element naming convention	131
Figure 6-4: Welded joint under a) tensile load; and b) bending load	133
Figure 6-5: Mesh convergence study of welded joint under tensile loading	134
Figure 6-6: Mesh distribution and stress contour plot of joint under tensile loading	135
Figure 6-7: Mesh distribution and stress contour plot of joint under bending loading	135
Figure 6-8: Abaqus scripting flow chart (Abaqus 2010a).....	137
Figure 6-9: Model generation of butt welded joint in Abaqus (Agarwal, 2011)	139
Figure 6-10: Mesh distribution and stress contours of case-1	141
Figure 6-11: Mesh distribution and stress contours of case-2	141
Figure 6-12: Relation between weld toe radius and stress concentration factor (SCF)	145
Figure 6-13: Relation between weld toe angle and stress concentration factor (SCF)	145
Figure 6-14: Relation between upper, lower reinforcement and stress concentration factor (SCF)	146
Figure 6-15: Relation between weld toe radius and stress concentration factor (SCF)	148
Figure 6-16: Relation between weld toe angle and stress concentration factor (SCF)	148
Figure 6-17: Relation between upper, lower reinforcement and stress concentration factor (SCF)	149
Figure 7-1: Layout of experimental work	151
Figure 7-2: Location of test specimens for a butt weld in plate	152
Figure 7-3: Geometry of tensile test specimen (EN ISO 6892-1:2009).....	153
Figure 7-4: Instron (3369) tensile testing machine	154
Figure 7-5: Mechanical behaviour of steel S355JR.....	155
Figure 7-6: Microhardness machine Buehler 1600-6100 (left), and close up of specimen (right).....	156
Figure 7-7: Hardness profile of butt welded joint	157
Figure 7-8: Dimension of butt welded joints: weld height h , weld width w , weld toe radius ρ , and weld toe angle α	158
Figure 7-9: Equipment used in geometrical estimation of welded joints.....	159
Figure 7-10: Dental moulding cement procedure	159
Figure 7-11: Weld joint parameter by using dental moulding	160

Figure 7-12: Fatigue test specimen.....	162
Figure 7-13: Fatigue testing machine	163
Figure 7-14: Characteristic fatigue strength S-N curves.	164
Figure 7-15: Examples of failure specimens.....	165
Figure 7-16: Examples of fatigue failure stages	166
Figure 8-1: A two beam structure with a load F.	169
Figure 8-2: Pareto front of the two-bar truss problem.....	170
Figure 8-3: Input variable-node Settings, the variable can be set as constant or variable.	171
Figure 8-4: Settings for the Input file-node.	171
Figure 8-5: Input file-node editor.	172
Figure 8-6: DOS-node properties, at the button Edit DOS Batch Script commands are written.....	173
Figure 8-7: Abaqus-node properties.	174
Figure 8-8: Odb-file Introspection at the Abaqus-node.	175
Figure 8-9: Objective-node properties.	176
Figure 8-10: Pareto front of the two-bar truss problem.....	177
Figure 8-11: Three Pareto optimal solutions provided by modeFRONTIER.	177
Figure 8-12: Dimension of the plate.....	178
Figure 8-13: modeFRONTIER workflow for plate with hole problem	179
Figure 8-14: Three- case study according to the mesh size	180
Figure 8-15: Pareto frontier of case 1 with (1117 elements)	181
Figure 8-16: Pareto frontier of case 2 with (2253 elements)	181
Figure 8-17: Pareto frontier of case 3 with (3113 elements).	182
Figure 8-18: Geometry of welded joint.....	183
Figure 8-19: Workflow modeFRONTIER of welded joint problem.....	185
Figure 8-20: Constraint node setting.....	186
Figure 8-21: Tensile stress vs. bending stress.....	188
Figure 8-22: Representation of Pareto front with different levels of constraints.	189
Figure 8-23: The multi-objective robust design optimization work flow.	190
Figure 8-24: The scheduler node properties.	191
Figure 8-25: Radar chart for Case-1	195
Figure 8-26: Radar chart for Case-2	195
Figure 8-27: Radar chart for Case-3	196

List of Tables

Table 2-1: A brief history of welding	23
Table 3-1: Basic concepts of Simulated Annealing.....	50
Table 3-2: Nomenclature of Particle Swarm Optimisation	52
Table 4-1: Results of comparison between Circumscription and Pair-wise Metrics	75
Table 4-2: Optimisation routine output for 16 trial points generated by Sobol.....	82
Table 4-3: Performance measures of MOGA-II, ARMOGA, NSGA-II, MOSA, FMOGA-II and MOPSO algorithms for lifting arm problem showing the values of Hit-Rate (HR%) Pair-Wise (PW) and Circumscription (CM) Metrics.....	89
Table 4-4: Performance measures of MOGA-II, ARMOGA, NSGA-II, FMOGA-II, MOSA and MOPSO for SCH and FON problems showing the values of Hit-Rate (HR %) and Circumscription (CM) Metric	96
Table 5-1: General parameters of tank problem	105
Table 5-2: Results of tank problem	106
Table 5-3: Traditional optimisation of lifting arm problem parameters	112
Table 5-4: Robust optimisation of lifting arm problem parameters	112
Table 5-5: Approximation of Pareto front and constraint lines	112
Table 5-6: Robust optimisation results for the first and second approach	117
Table 5-7: Robust optimisation results for the third approach	118
Table 6-1: Geometrical parameters of butt welded joint.....	130
Table 6-2: Material properties of steel S355JR	131
Table 6-3: Equation for extracting x and y coordinates of welded joint geometry	139
Table 6-4: Range of geometric parameters	142
Table 6-5: Geometric parameters of welded joint under tensile load	144
Table 6-6: Geometric parameters of welded joint under bending load	147
Table 7-1: Dimension of 3 tensile test specimens (EN ISO 6892-1:2009).....	154
Table 7-2: Mechanical properties of steel S355JR	155
Table 7-3: Micro hardness results of butt welded joint	157
Table 7-4: Estimation results of 11 welded joints	161
Table 7-5: Fatigue test results	164
Table 8-1: Input and output variables name in modeFRONTIER software	185
Table 8-2: Design parameters problem	187

Table 8-3: Initial and optimized design parameters	187
Table 8-4: Pareto front with constraint.....	189
Table 8-5: Input design parameters of three cases.....	193
Table 8-6: Robust design parameters of three cases.....	194

List of abbreviations and symbols

ANNs	Artificial neural networks
ARMOGA	Adaptive range Multi-objective Genetic Algorithm
ASTM	American Society for Testing and Materials
AWS	American welding society
BS	British Standard
C	Crack growth rate coefficient
CM	Circumscription metric
DoE	Design of experiment
E	Modulus of elasticity
FCG	Fatigue crack growth
FCG	Fatigue crack growth
FE	Finite element
FEA	Finite element analysis
FEM	Finite element method
FMOGA-II	Fast Multi-objective Genetic Algorithm
FSW	Friction stir welding
GA	Genetic Algorithm
GMA	Gas metal arc welding
GTAW	Gas tungsten arc welding
HAZ	Heat affected zone
HV, HB	Vickers and Brinell hardness
IIW	International institute of welding
LEFM	Linear elastic fracture mechanics
MOGA-II	Multi-objective Genetic Algorithm
MOPSO	Multi-objective Particle Swarm Optimisation
MOSA	Multi-objective Simulated Annealing
PM	Parent material
PW	Pair wise metric
R	Stress ratio
SCF	Stress concentration factor
STD	Standard deviation

t	Plate thickness
TIG	Tungsten inert gas
WM	Weld metal
α	Weld toe angle
ν	Poisson's ration
ρ	Weld toe radius
σ_a	Stress amplitude
σ_m	Mean stress
σ_{max}	Maximum stress
σ_{min}	Minimum stress
σ_y	Yield stress
ΔK	Range of stress intensity factor
$\Delta\sigma$	Stress range

Chapter 1

Introduction

Chapter One: Introduction

1.1 Motivation for the Research

Welding is the most dominant joining method involved in the manufacturing process of all components of structures, such as pressure vessels for nuclear power plants, oil and gas pipelines, submarines, and rocket engines in the aerospace industry; it is also the main source of failure occurring in components.

The assessment of welded joints is the most important problem seen to arise in the industrial field because the weld is the governing factor in the life expectancy of structures. Welded joints are the source of weakness in structures, and therefore need to be fully investigated in order to improve the service life of the structures.

The critical problem in the assessment of welded joints is the difficulty associated with specifying the weld geometry in a manner that is sufficiently simple for industrial use but sufficiently precise for analysis.

Importantly, the failure of welded joints can have several consequences, both human and economical. One such example, which occurred on March 15, 1979, involved the British motor tanker *Kurdistan*, which broke in two, en route from Nova Scotia to Quebec. The defective butt weld in the port bilge keel was the origin of the crack (see Figure 1-1). There was a lack of penetration in the butt weld and where the bulb plate overlapped the underside of the ground bar.

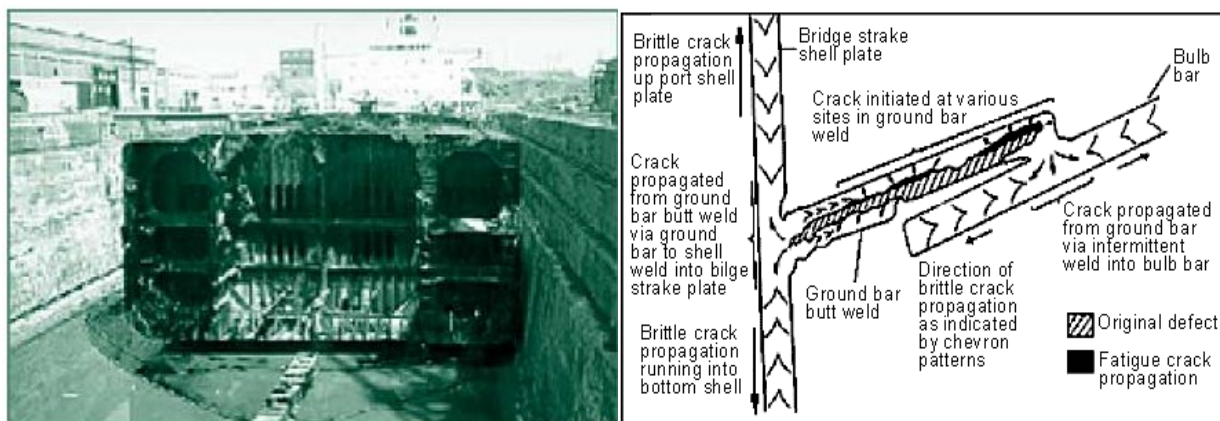


Figure 1-1: M V Kurdistan tank failure (Garwood, 1997).

The collapse of the off-shore platform Alexander Kielland was the second famous accident, as shown in Figure 1-2. The hydrophone holder welded into tubular bracing- which is known to represent the important structural member of Alexander Kielland- is one example of welded pipe penetration where fatigue failure became catastrophic for the platform.

The investigation showed that a fatigue crack had propagated from the double fillet weld near the hydrophone mounted to the tubular bracing (see Figure 1-3). Lotsberg, I. (2004)



Figure 1-2: Disaster of the Alexander Kielland platform (Kristoffer, 2012).

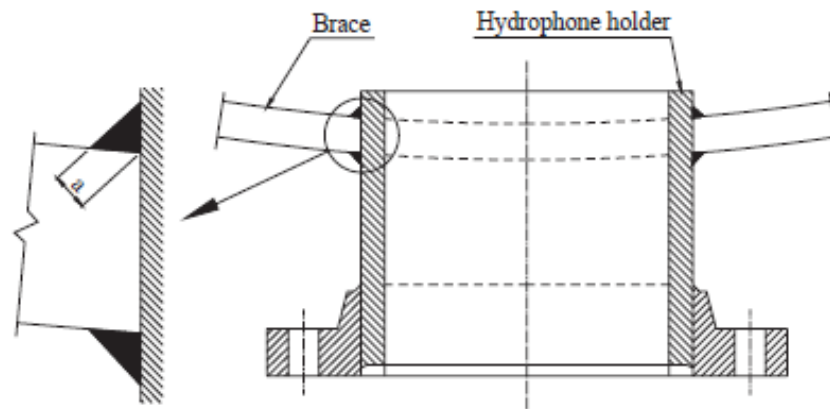


Figure 1-3: Hydrophone holder welded into tubular bracing of the Alexander Kielland platform (Lotsberg, 2004)

1.2 Aim of Study

Product performance is, now more than ever, a critical requirement for success in manufacturing and production; however, significant uncertainty exists in geometrical parameters of welded

joints, such as weld toe radius, weld toe angle, upper and lower reinforcement and plate thickness which are critical to system performance.

The aim of this project is to create a robust design of welded joints which is good in terms of performance, and which is insensitive to design parameter variations under different types of loads (tensile and bending).

A few experimental results and the finite element model gave further insight into the factors affecting the fatigue life of welded joints.

1.3 Objective

The aim of this work was to produce a methodology for the robust design of butt welded joints. This has been broken down into the following objectives:

- To determine the current state-of-the-art weld geometry optimisation approaches.
- To identify the most appropriate tool for multi-criteria structural optimisation.
- To implement a method of embedding finite element analysis within an optimisation tool.
- To devise a method of identifying robust designs from within the Pareto set for multi-criteria problems.

1.4 Contribution to Knowledge

The major contributions to knowledge described in this thesis are:

- The development of a new methodology for selecting the most robust designs from within the Pareto set so as to ensure the design is both robust and optimum
- Evidence to indicate that the Circumscription Metric (CM) is superior to the Pair Wise metric (PW), particularly for optimisation carried out using genetic algorithms.

1.5 Structure of the Thesis

This thesis presents the detailed results of the author's work over a PhD research programme, which was aimed towards achieving the objective of the research. It includes a detailed literature review of the research project, a numerical simulation and an experimental study. The thesis is divided into the following nine chapters.

Chapter One: Presents the motivation behind the research problem, along with the objectives and structure of the thesis.

Chapter Two: Provides a general background to welding, welded joint parameters, optimisation of welded joints, fatigue, fatigue loading and the fatigue of welded joints

Chapter Three: Presents a detailed description of the optimisation method, which includes introduction, historical development and engineering applications. It will also describe the mathematical formulation of single and multi-criteria optimisation. This chapter will present different methods of optimisation, which includes classical methods and population-based evolutionary algorithms. Additionally, a mathematical formulation and description of a robust optimisation method will be presented in this chapter.

Chapter Four: Presents the optimisation software with initial population generation methods. Different assessment metrics are given in order to evaluate the performance of the optimisation algorithms that exist in the optimisation software. Two different types of problem are evaluated in this chapter with genetic algorithms.

Chapter Five: Introduces the philosophy of the robust optimisation with two application problems, the first one with single objective optimisation and the second with multi-criteria robust optimisation.

Chapter Six: Presents the finite element parametric study carried out to investigate the effects of several geometrical parameters on stress concentration factors in welded joints. A Python script of a welded joint model is developed in this chapter.

Chapter Seven: Introduces the experimental work, and comprises the monotonic tensile test, hardness test, geometrical estimation of the welded joints and the fatigue test.

Chapter Eight: The first part of this chapter presents the integration of the ABAQUS software and optimisation software with two detailed examples. The second part includes the traditional and robust multi-criteria optimisation of welded joints based on the model developed in Chapter Six and the values of geometrical parameters identified in Chapter Seven.

Chapter Nine: This chapter presents the conclusions of the research and makes suggestions for further work.

Chapter 2



Welding and Fatigue

Chapter Two: welding and Fatigue

2.1 Historical Background on Welding

Welding refers to the processes whereby metal parts are joined together. It is an essential process used in the manufacture of many different products made of metals. In fact, welding is well-known for its cost efficiency and long lasting results. Although welding can be used for all kinds of metals, there are some metals that are difficult to weld. This difficulty depends on the natural properties of metal materials. However, the process of welding has increasingly grown and developed because of its wide range of effective applications in industry (Cray, 1979). New welding methods have been invented while the earlier processes are still employed. Iron forge welding and diffusion bonding, for instance, are still reliable techniques of welding. In addition, the cold pressure welding method, which is currently applied in some aspects of modern industry, dates from the Bronze Age. A brief history of welding is indicated in Table 2-1.

Table 2-1: A brief history of welding

In the mid-1800	The foundation of modern welding was laid by Sir Humphrey Davies of England with two discoveries: 1- Acetylene 2- The production of an arc between two electrodes of carbon
From 1877 to 1903	A great number of inventions and discoveries, such as gas welding and cutting, arc welding, resistance welding
1905	In 1905, Pirani was first to use electron beams for fusion tests with metals (Schiller S. 1977). This technology used a beam of electrons that are accelerated by a high voltage and can so be used as a tool for treatment of materials such as in welding
1916	Oxyacetylene appeared as a more efficient welding method for copper, aluminium and steel. Howard B. C. (1998)
In 1920	In 1920, gas-shielded metal-arc welding (GMAW) was introduced to allow continuous welding with its inherently greater productivity, but was limited, especially in positional welding, by its lack of slag. (Nasir A. 2005).

1930	First appearance of gas tungsten arc welding (GTAW) in the USA in the late 1930s, where it was used for welding aluminium airframes. It was an extension of the carbon arc process, with tungsten replacing the carbon electrode. (Nasir A. 2005).
From 1943	The development of welding methods took on a quicker pace in the industries where metal products are manufactured (Lancaster, 1987).
1950	In the early 1950s, the first demonstration of ultrasonic metal welding process. It was found that ultrasonic vibrations were capable of creating a weld in metal parts without the need for melting the base metals. Howard B. C. (1998)
By 1953	Carbon dioxide was used by Lyubavski and Novoshilov to provide a shield gas for consumable electrodes welding. This method was widely used but it required high currents of electricity
1954-1957	Attempts to refine CO ₂ welding did not stop and new advances were made. In addition to the gas emitted from the wire, an external shield of gas was applied to yield a better method of welding called the dual shield. This method was originally invented by Bernard in 1954, but it was not used until 1957 that the method gained publicity.
1960	In 1960, DuPont filed the first internationally recognized patent on explosion welding technology (EXW), Cowan G. R. <i>et al</i> (1964). There was extensive research During the ensuing 20 years concerning this technology. In 1962 DuPont commercialized the explosion cladding industry, with the first major application being production of tri-layer coinage for the US government.
1970	Thus laser welding was announced almost simultaneously by three different suppliers of laser equipment who were seeking to expand their markets (Locke E. V. <i>et al.</i> 1972). Most laser beam welding is conducted by the output of either the neodymium-doped yttrium aluminium garnet (Nd:YAG) laser or carbon dioxide laser or both of these lasers, depending on the electrical excitation circuitry, can emit their output either continuously, as a single pulse, or as a repetitive series of pulses.

1991	In 1991 friction stir welding (FSW) was invented by Wayne Thomas at (The Welding Institute), and the first patent applications were filed in the UK in December 1991, and found high-quality application all over the world.
	During the 1990s and continuing today, rapid changes are evolving in the welding industry as engineers devise more advanced filler metal formulas to improve arc performance and weld, it has shown that advancements are inevitable and will continue, such as state-of-the-art electrodes, exotic multiple gas mixes, hybrid processes, robotic welding and onboard computers. Ibrahim, K. (2007)

2.2 Welded Joint Parameters

The main factors that affect and control the fatigue strength of welded joints are weld profiles and weld imperfections.

2.2.1 Geometrical parameters of welded joint

The main parts of butt and fillet welded joints are indicated in Figure 2.1.

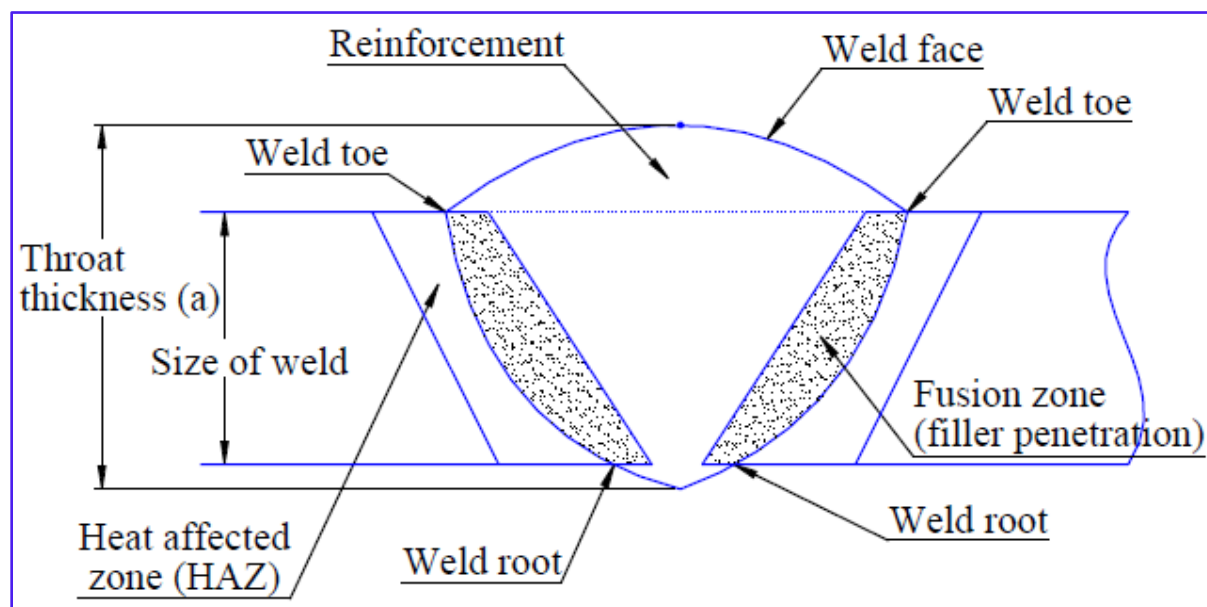


Figure 2-1: Geometry of a butt weld (American Welding Society, AWS A3.0M/A 3.0:2010 and British Standard Institution BS EN ISO 9692-1:2013)

The junction between the base metal and the face of the weld is called the weld toe, while the intersection between the parent metal and the back of the weld is called the weld root.

Ideally, the straight line between the weld toes forms the face of the weld. The throat thickness 'a' is the minimum distance between the weld root and the weld face. The portion of weld from the root to the toe is known as the leg or size of the weld

The heat affected zone (HAZ) is the part of the base metal which is below melting point but undergoes a high rate of heating and cooling during the welding process. This part can become brittle because it is subject to a hardening treatment.

The heat affected zone (HAZ) and fusion zone of welded joints can show very different material properties from base metal as well as between themselves. For example, the heat affected zone can exhibit a heat treated structure involving re-crystallisation, phase transform and grain growth while the fusion zone exhibits a typical cast structure.

2.2.2 Weld imperfections

Figure 2.2 shows a sketch of a cross section through a butt welded joint. Some defects which determine the weld quality are indicated. These and other defects can be summarised as follows (Shinagawa and Ku, 2011).

- *Cracks*: This imperfection may arise from the effect of stresses or cooling. Due to their geometry, which produces a high stress concentration at the crack tip, cracks are the most critical imperfections. They can be classified according to their location (longitudinal and transversal crack), or according to their nature (hot and cold crack).
- *Porosities*: May be either due to shrinkage during solidification or gas cavities due to entrapment of gas.
- *A lack of fusion and penetration*: This imperfection is the result of improper penetration of the weld materials, improper welding techniques or improper joint design.
- *Imperfect dimension and shape*: The first form of this type of imperfection is the undercut which is associated with either excessive welding currents or improper welding

techniques, or both. It is located at the junction of the base and weld metals. Other similar types of imperfection are: misalignment, irregular width overlap, excess penetration and excess weld metal.

- *Solid inclusions*: An imperfection caused by a solid foreign substance entrapped in the weld metal (flux, slag, oxide and metallic inclusions).
- *Miscellaneous imperfections*: Excessive reinforcement, concave and convex fillet weld, undercut, under fill, overlap and incomplete fusion.

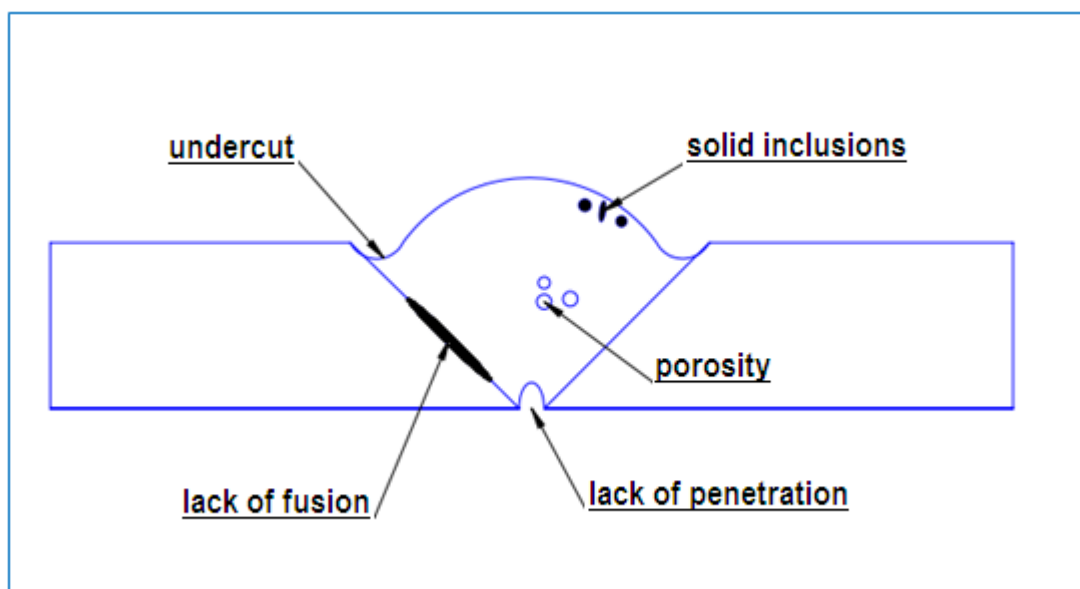


Figure 2-2: Defects of a butt welded joint

2.3 Optimisation of Welded Joints

2.3.1 Effects of welding parameters on metal properties

Each type of metal has certain mechanical properties. Such properties are responsible for giving the metal materials different levels of hardness, resistance to wear, etc. In the welding process, the amount of heat applied to the Heat Affected Zone (HAZ) should be, therefore, proportional to the capability of metals to handle such stress. However, the mechanical properties of the weld and HAZ differ from the general properties of the base metal. This difference may impact upon the quality of the weld.

Cracks often occur in metals during the welding process. Therefore, Scotti and Albuquerque Rosa (1997) conducted research to find out the best ways to enhance metal properties before welding. A number of tests were performed on the metals welded through the automatic flux cored arc method. The experiments carried out statistically measured the effects of the welding parameters on metal properties. The findings revealed that the size of the oscillation (transverse movement of welding torch) amplitude and the amount of preheating are the prime factors behind crack control. The probability of cracking can be minimised by maximising pre-heating and reducing oscillation amplitude. Moreover, the experiments found that the problem of weld cracking can be controlled without any negative effect on hardness.

Plastically deformed steel bars produced by a forging process was the focus of a study made by Sahin and Erol Akata (2003) to investigate the effect of welding parameters of friction welding. In their research, steel bars with similar and dissimilar diameters were welded while welding parameters were measured statistically. It was observed that there is an inverse relationship between the tensile strength of the weld and its width. Plastic deformation of steel bars did not affect the welding zone because of the high temperature in the welding process. The research concluded that friction welding is one of the most appropriate methods to weld plastically deformed steels. However, to guarantee successful results, parameters of the welding process including friction pressure, pressure time, upset pressure, upset time and rotational speed need to be carefully controlled.

The mechanical properties and geometry of titanium alloy Ti6Al4V weld bead created using a CO₂ laser beam were examined by Caizzo, Curcio and Daurelio (2004). Additionally, Helium and Argon gas shields with different gas pressures, weld covering gas nozzles and characteristics of the weld joint were employed in the experiment so as to investigate the effects of such gases on the properties of the welded materials. No significant difference in weld bead geometry was observed. However, variations of micro-hardness were detected between the base metal and the Heat Affected Zone: the micro-hardness value at the HAZ was twice its counterpart at the base metal. Furthermore, the researchers conducted tensile tests on Bead on Plate (BoP) and butt welds to investigate the effect of edge alignment prior to the welding process. The results showed that there were no differences between the base metal and BoP with regard to strength. However, the stress-strain diagram revealed a failure in the butt welds starting from the bead foot regardless of different thicknesses. This weakness of the butt joints is concluded as a result of

two factors: the misalignment of the edges and an increased air gap. Helium gas should be used to protect the lower sides of the joints.

Three different back-propagation neural network models were investigated in a study conducted by Sterjovski, Nolan and Carpenter (2005) to predict (i) hot ductility (reduction of area) and strength (ii) toughness and (iii) hardness of the heat affected zone of steel parts after welding. The models proved successful in anticipating the mechanical properties of the welds. In doing so, the models can save the costs and time required for conducting experiments dedicated to investigating the parameters that produce an optimised welding process.

Hurtier *et al.* (2006) took the investigation of friction stir welding (FSW) methods further but with a focus on the mechanical and thermal history of the welded metal. The micro-hardness of the welded joints was examined along with the temperatures in the welding process. The study concluded that the quality of the welds can be increased if the average temperature of weld zone is reduced. In addition, the properties of the fusion area should be studied prior to welding in order to identify the proper parameters for better micro-hardness. The presence of oxide in the weld weakens its structure; additionally this model is also used to indicate the presence of a weakened zone in the weld.

Friction welding was the prime concern of Yoon, Kong and Kim (2006) who made attempts to optimise the process. The researchers used reduced activation ferritic steel (JLF-1), stainless steel (SUS304) and acoustic emission (AE) to evaluate the real-time weld quality. In friction welding a strong plastic deformation occurs until friction welding process completion, and then prominence of acoustic emission occurs during formation of flash, and it was confirmed that real time cumulative total AE counts could be measured. The properties of models were tested against the welding technique parameters and acoustic emission was confirmed as a successful tool for evaluating the quality of the weld.

From the literature, there were many studies carried out to using Genetic algorithms for the study of effect of welding process parameters either on the welded joint strength or on weld bead geometry for various welding process these studies are discussed in the following sections:

Kim, Rhee and Park (2002) illustrate the use of a genetic algorithm and a response surface methodology in the modelling and optimisation of gas metal arc (GMA) welding process parameters. Welding processes are complex procedures whose quality outcomes are mostly

based on a trial and error method. The trial and error method adversely affects the production quality and thereby affects an institution and organisation in terms of costs and time. They argue that response surface methodology and genetic algorithms can be utilised and employed to optimise welding processes and procedures. The researchers utilised genetic algorithms to determine near optimal conditions to minimise a number of preliminary experiments for welding processes while the response surface methodology was utilised and employed to determine the optimal conditions during the welding process. The optimum results were judged based on a desirability function. In this study the desirability function approach was used to optimize three bead geometry variables (bead width, bead height and penetration). In other words, they used a weighted criteria method, a technique which has some fundamental flaws as explained in the next chapter.

Ersel Canyurt (2005) attempted to predict the strength value of welds by developing a Genetic Algorithm Welding Strength Estimation Model. Experiments were carried out on brass joints with five different parameters (gap between the parts, torch angle, quantity of shielding gases, pulse frequencies and the electrode tip angle during welding). The resultant data indicated that changing the distance between welded parts from 0 to 0.5 mm leads to a 4.4 times decrease in the strength value of the welded parts. Moreover, the quantity of gas used in the gas shield and the pulse frequencies contribute to the tensile strength of the welds. The developed models were validated with experimental data. However, this study was restricted to the welding of one particular metal alloy

Two attempt have been made to correlate welding process parameters with bead geometry the first one by Vasudevan et al. (2007) and the second one by Sudhakaran et al. (2011).

In the first study, The development of methodology for optimizing a tungsten inert gas (TIG) welding process parameters (current, voltage, torch speed, and arc gap) using genetic algorithm to achieve the target weld bead geometry (bead width, depth of penetration and reinforcement height) for welding 304LN and 316LN stainless steel was done. Whilst in the second study, the direct effect and interactive effect of parameters like welding current, welding speed, shielding gas flow rate and welding gun angle on bead dimension has been investigated during this study. They compared the results from this model with experimental results and they found that the developed model was able to predict bead dimension with reasonable accuracy.

From these studies, it can be inferred that: all the previous studies investigated the effect of the welding process either on welded joint strength or welded joint geometry and in many cases, this has been aided by GA's with experimental data used to validate the results. However, no work has been reported studying the combined effect of all geometrical parameters on the strength of welded joints. In addition, within this field there has been no comparison of the different GA's available

2.3.2 Effects of geometrical parameters on welded joints

Another research study that sheds light on the reasons as to why welded joints can be weak and of poor quality was conducted and carried out by Fricke in 2003. He argued that welded joints are not perfect and of high quality because they contain pores, cavities and inclusions. These factors contribute to the weakness of a welded joint. In addition, Fricke noted that the shape of a welded joint on high stress concentration features. This makes welded joints susceptible to fatigue failure.

According to Benyounis and Olabi (2008), the quality of a welded joint is positively related to the welding input parameters. This implies and signifies the materials and components that are utilised and employed during a welding process will in effect determine the strength and quality of a welded joint. In addition, they argue that a welding joint is defined and described in terms of its properties, i.e. the welded joint's mechanical and physical properties, distortion and weld-bead geometry.

2.4 Introduction to Fatigue

Fatigue is defined as:

“The process of progressive localized permanent structural change occurring in a material subjected to conditions that produce fluctuating stresses and strains at some point or points and that may culminate in cracks or complete fracture after a sufficient number of fluctuations”

(ASTM E-1823, 2000).

When a material is exposed to cyclic stress below its ultimate tensile strength, localised hardening or softening occurs due to plastic deformation. The location(s) of plastic deformation might be at points of stress concentration or even in the absence of a stress raiser as in the case of persistent slip bands where crack initiation could be due to extrusion formation. After the application of a number of cyclic stresses, cracks can appear in a structural member due to localised structural change. In this process, fatigue loading is the term used to describe the

applied loading, and fatigue failure is used to describe the resultant fracture. The fatigue life of a component is the term used to describe the amount of time during which the member can endure the cyclic load before fracture occurs. This fatigue life might also refer to the number of load cycles used to cause fracture.

Fatigue failure of metals occurs in almost all engineering fields. It has been assessed that up to ninety percent of common mechanical failures are a result of fatigue failure (Stephens, Fatemi, Stephens & Fuchs, 2001). It is also believed that such failures occur unexpectedly. Fatigue failure of the metal structures occur in every field of engineering such as the electrical, thermal, mechanical, civil, aeronautical, biomedical, chemical and nuclear fields.

Furthermore, fatigue failures can be found in problems of simpler items such as paper clips that crack after a number of bending actions; door springs that crack after repeated moving actions; electric light bulbs that stop working due to failure in the electric circuit; tennis racquets, tooth brushes, etc. Additionally, fatigue failure can be observed in complex components and structures like steering linkage of automobiles, connecting rods of engines, propeller shafts of ships, landing gears and fuselages of airplanes and even in implanted organs in human bodies, vehicles, vessels, airplanes, ships and humans are subject to this kind of failure

In fact, cracks and fractures represent an original threat to engineering structures. A crack becomes a serious threat leading to fatigue failure when its size is bigger than a critical size. Despite the fact that not all cracks are critical, structural failure can still occur when subcritical cracks are developed by fatigue loading. This development of cracks is referred to as fatigue crack growth (FCG). Since in some cases it is difficult to stop the growth or development of subcritical cracks in almost all structures, a concept of crack sensitivity (damage tolerant design) has been developed for structures subject to complete fatigue failure.

The concept of Fatigue Crack Growth has been utilised by design engineers in order to produce designs for long fatigue lives.

2.5 Fatigue Failure Stage

A metal goes through three main stages before finally reaching complete failure. These stages are called the stages of fatigue failure. The first stage is called the crack nucleation. This stage or process is essentially influenced by voids or inclusions. It is believed that local stress and strain concentrations are the factors that control this process of nucleation which is attributed to cyclic

slip. The second stage in a metal fatigue life is called micro-crack growth. This stage is used to describe the development of slip bands, voids or inclusions in a metal up to 20 μm , so this stage is usually known as a small-crack growth process. There is much interest in this stage which is concerned with the behaviour of small cracks because it has been found by Schijve, (1979) that small cracks of 100 μm consume from 60 to 80 percent of fatigue life in a pure and polished metal's surface and studies by Newman (Newman & Edwards, 1988) and Newman *et al.* (Newman, Wu, Venneri & Li, 1994) showed that up to 90 percent of a metal's fatigue life is consumed in crack growth from 10 μm to failure. The final stage that precedes fatigue failure is the macro-crack growth. In the regions of macro-crack growth, it is usually easy to predict the growth of fracture leading to fatigue failure.

The three general phases of fatigue life are shown in Figure 2.3. These stages in the figure are representative of the kind of fracture that occurs on the surface of metals. The application of repeated loads is noticeably the one thing in common between the first and the second stages. The nature of the applied load could be either mechanical, as in the crank shaft of a combustion engine, thermal, as in heat exchangers and boilers, or it could be represented in repeated pressures as in an aircraft's interior, fuselage or wing fixations.

Because of the great significance of the first two stages and their greater impact on the development of fatigue fracture, the total sums of all cycles that occur in these stages represents the fatigue life of a member.

The fatigue life of a component, i.e. the total number to failure, is the sum of cycles at the first and second stages:

$$N_f = N_i + N_p \quad (2.1)$$

Where

N_f = The total number of cycles leading to eventual fracture.

N_i = The total number of cycles responsible for the initiation of a crack.

N_p = The total number of cycles responsible for the propagation of a crack.

It has been found that the value of N_i can be high when the stresses are mostly elastic in high cycle fatigue (Schijve, 1979). In low cycle fatigue, the value of the metal's yield strength is lower

than the applied stress levels and bulk plasticity is present. N_p becomes dominant and N_i goes down when the stress level is increased.

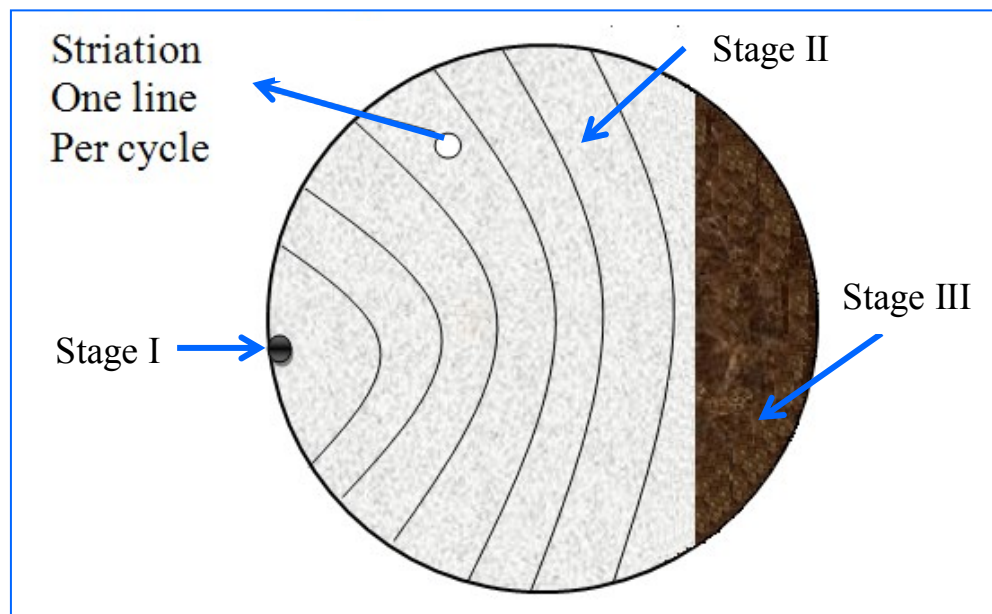


Figure 2-3: Fatigue stages on the fracture surface

2.6 Fatigue Crack Growth Rate

Crack growth rate is defined as crack extension per cycle (da/dN). Consider a crack that is propagating in the presence of a constant amplitude cyclic stress intensity factor (Δk), Where $\Delta k = K_{max} - K_{min}$. The rate of fatigue crack growth per cycle (da/dN) is governed by stress intensity factor range (ΔK), a plot of $\log (da/dN)$ versus $\log (\Delta K)$ is shown in Figure 2.4.

As shown in Figure 2.4, the growth process of a fatigue crack has three general stages represented in the figure by three different areas that divide the growth curve (Molent, Jones, Barter & Pitt, 2006). Each of these areas or regions has its own properties and characteristics. The first region represents the threshold value of ΔK_{th} that is considered the initial development of a crack. All values that fall below ΔK_{th} are considered as non-notable growth and as non-propagating cracks. It is believed that this region has a strong link to the crack slow development of a crack under low ranges of stress. It is also argued that this region is responsible for a great proportion of a metal's fatigue life.

The second area or region is characterised by stability of crack development. This region has been extensively studied for its technical significance and it is usually called the Paris region

(Huang & Moan, 2007). The importance of this region lies in the fact that it is the region where the Paris law (Paris & Erdogan, 1960) can be applied. This law defines a linear relationship between $\log |da/dN|$ and $\log |\Delta K|$:

$$da/dN = C(\Delta K)^m \quad (2.2)$$

Where m is the slope of the line generated by the equation and the coefficient C is generated when the straight line is extended to $\Delta K = 1 \text{ MPa } \sqrt{\text{m}}$. The values of C and m are constant. This stage lasts for a certain period of time until the third stage initiates and the final fatigue failure occurs in a material i.e. K_{IC} is reached.

The development of the fatigue crack in the third region has its highest rates. Development continues rapidly towards a state of instability. This stage takes a small proportion of a material's fatigue life. The factor that predominately controls this region is the metal's K_{IC} (fracture toughness).

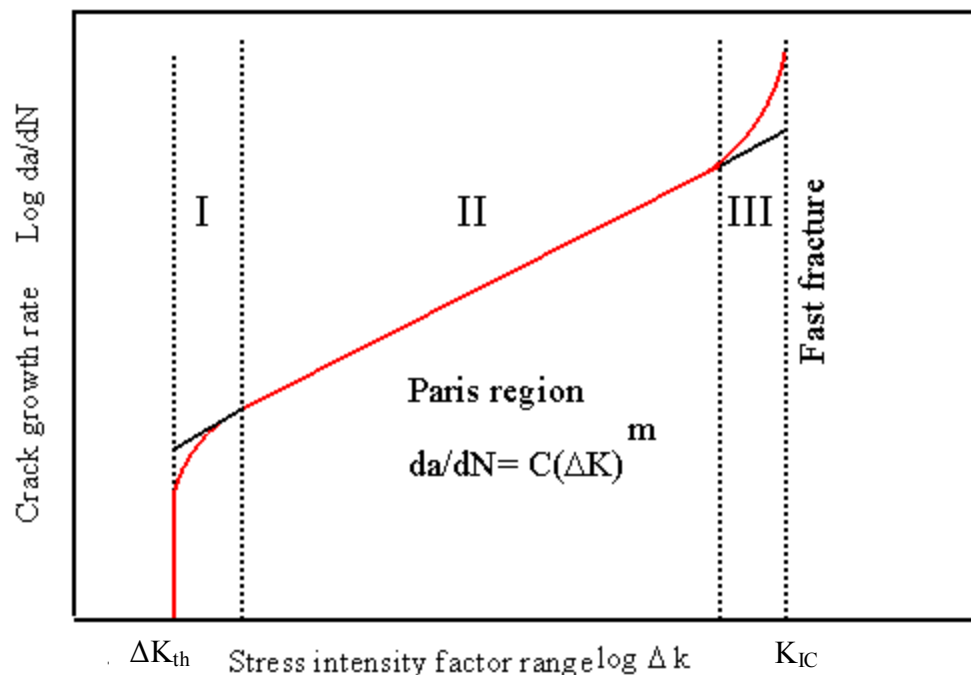


Figure 2-4: Typical relationship between the crack growth rate and the range of stress intensity factor.

2.7 Fatigue Loading

There are two types of load history in fatigue design. The first is the constant-amplitude cyclic load. This type of loading normally occurs in the laboratory during fatigue testing.

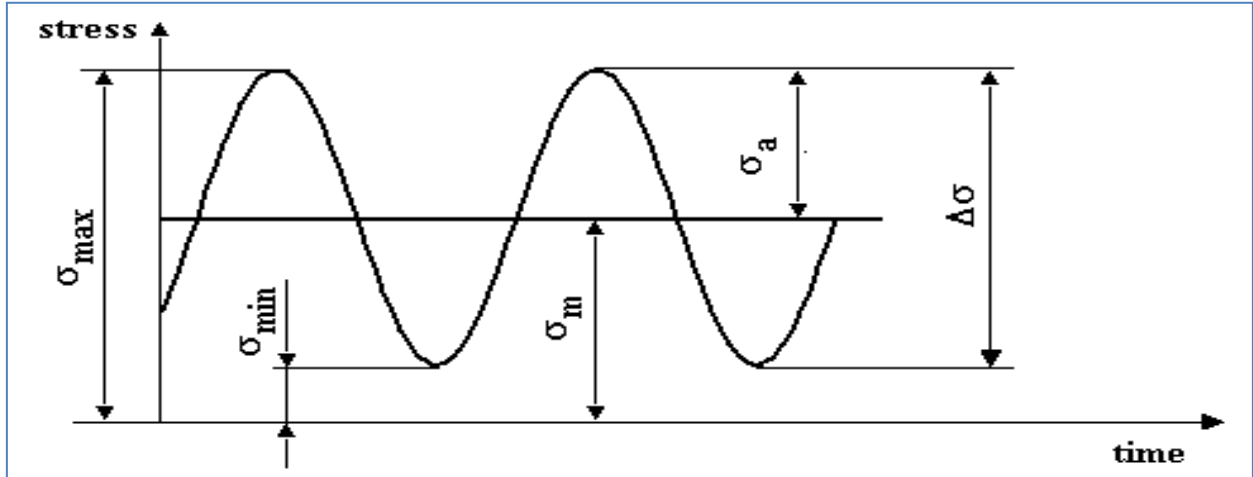


Figure 2-5: Nomenclature used in constant-amplitude loading (Wei & Pentti, 2003)

The main parameters of constant-amplitude loading are (Figure 2.5):

- Stress range: the difference between maximum and minimum stress

$$\Delta\sigma = \sigma_{max} - \sigma_{min} \quad (2.3)$$

- Stress amplitude: half the stress range

$$\sigma_a = \frac{\Delta\sigma}{2} = \frac{\sigma_{max} - \sigma_{min}}{2} \quad (2.4)$$

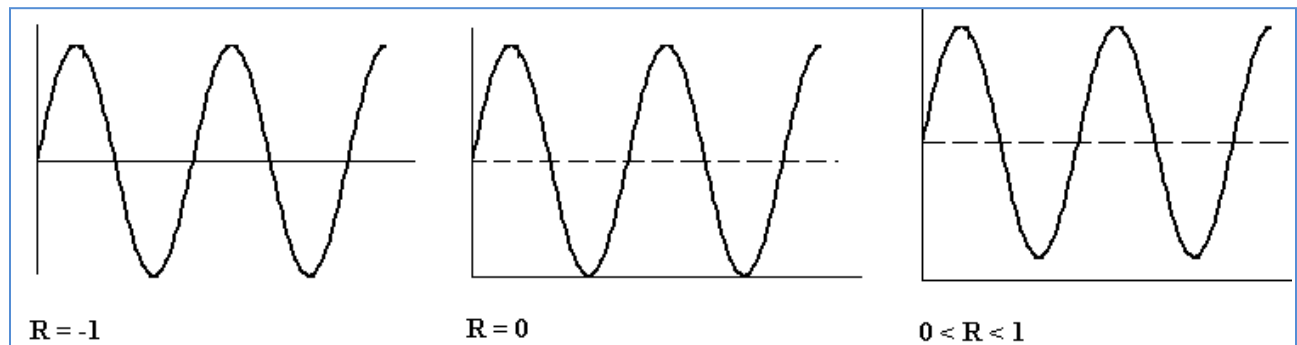
- Mean stress: the algebraic mean of maximum and minimum stress

$$\sigma_m = \frac{\sigma_{max} + \sigma_{min}}{2} \quad (2.5)$$

- Stress ratio: the relative magnitude of the minimum and maximum stress.

$$R = \frac{\sigma_{min}}{\sigma_{max}} \quad (2.6)$$

Values of R for different load cases are indicated in Figure 2.6



Fully reversed load

Zero to full tensile load

Tensile load

Figure 2-6: Comparison of R -ratios for different loading (Barsom & Rolfe, 1999).

The second type of loading is called variable-amplitude loading. This is more complex and it is difficult to represent with an analytical function (see Figure 2.7). According to the statistical data, 80% of structural fatigue failure is caused by this load Wei & Pentti, 2003. For example, rotating shaft, reciprocating springs, bearings, airplane structures and ships are mostly fractured under random loads. Actually there are several techniques used to idealised this load closer to laboratory load, such as short time Fourier transform and wavelet methods.

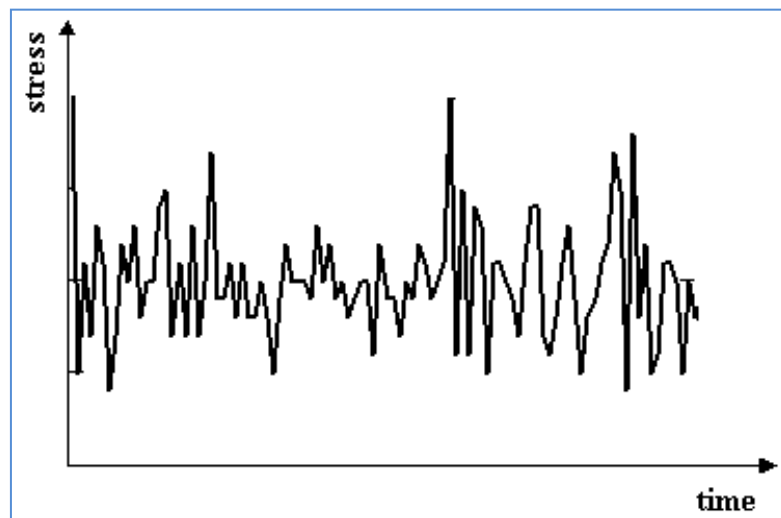


Figure 2-7: Variable-amplitude loading (Wei & Pentti, 2003)

2.8 Fatigue of Welded Joints

The durability and safety of welded structures become an important issue, because the failure of such systems may cause many injuries, environmental damage and financial losses.

The design of welded joints against fatigue is mainly based on different local approaches which are reviewed by Radaj (1996). The importance of the application of local approaches is justified by the fact that the fatigue process cannot be well defined by global stresses and it has a local character. The local approaches can be divided into the following three groups (Figure 2.8): the elastic structural stress or strain approach, the notch stress or strain approach and the fracture mechanics approach.

A literature review on the fatigue analysis of welded joints was conducted by Fricke (2003) and included mainly papers and books published during the past 10-15 years. An extensive selection of nominal stress, structural stress, notch stress, and notch strain and crack propagation approaches were included in this study.

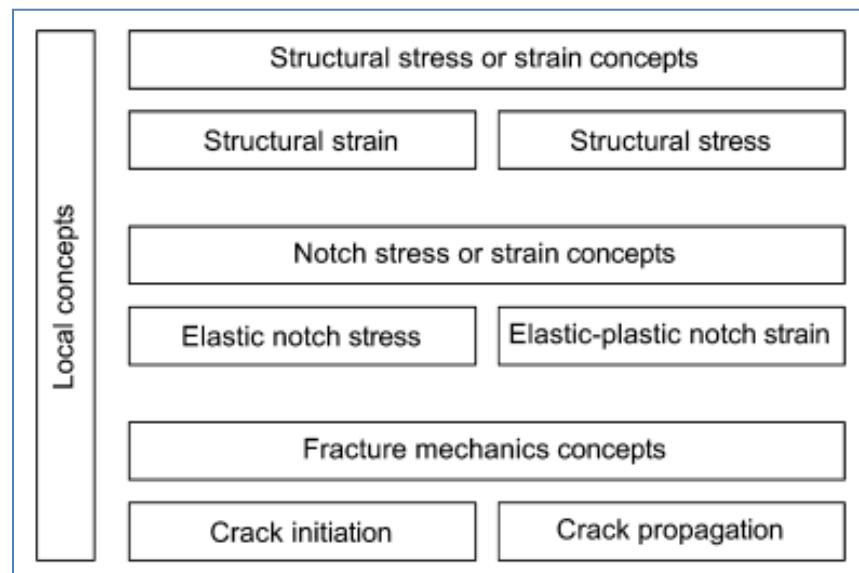


Figure 2-8: Local approaches for fatigue assessment of welded joints.

A detailed in-depth description of the fatigue assessment of welded joints by local approaches considering the state of the art research work is provided by Radaj, Sonsino and Fricke (2006).

Different formalisms have been examined by Boukharouba, Gilgert, and Pluvinage (1999) by calculating the stress concentration factor according to the weld toe radius and weld toe angle. A comparison between these results and those stemming from finite element methods has been

conducted. The results indicate that the stress concentration factor is related to weld geometry; this leads to a decrease in the fatigue life of welded joints which is more significant than the role played by residual stress.

Nguyen and Wahab (1998) developed a mathematical model to predict the fatigue behaviour of butt welded joints, taking into account various weld geometry parameters such as weld toe radius, weld toe under cut, plate thickness and weld toe angle by using the concept of linear elastic fracture mechanics (LEFM) and finite element analysis (FEA). They have found that the effect of weld toe undercut is very significant in comparison with the other butt weld geometry parameters.

In the study of Teng, Fung and Chang (2002) a finite element analysis (FEA) was conducted on butt welded joints to obtain the local stress and residual stress distribution, taking into account the geometrical parameters such as weld toe radius, weld flank angle and plate thickness. The results from this study indicate that modifying weld geometry by increasing the weld toe radius led to an improvement in the fatigue life of butt welded joints.

In order to evaluate the influence of the factors of stress concentration at both the weld toe and internal defects on the fatigue strength of the high strength steel MOMEX 600 CD for SSAB, a fatigue study was performed by Costa, Ferreira and Abreu (2010). Three different welding conditions were investigated in the course of this study. They concluded that the stress concentration at weld toe and initial defects are the main factors responsible for the fatigue strength of the welded joints.

Cerit, Kokumer and Genel (2010) investigated the stress distribution of welded joints with different values of reinforcement metal under uni-axial tension. A parametric two-dimensional finite element analysis was carried out. They found that a low reinforcement angle and weld toe radius cause higher values of stress concentration factors. Consequently, this value will reduce the load carrying capacity of these joints.

Chapetti and Jaureguizar (2011) attempted to predict the fatigue strength of welded joints by using a fracture mechanics approach. A comparison between finite element analyses of butt welded specimen and experimental results from the literature were carried out considering the influence of plate thickness and reinforcement angle. The comparison indicates that the reduction

in the reinforcement angle will reduce the scatter observed. This is due to the high stress gradients near the weld toe with small values of reinforcement angle (α), see Figure 2.9.

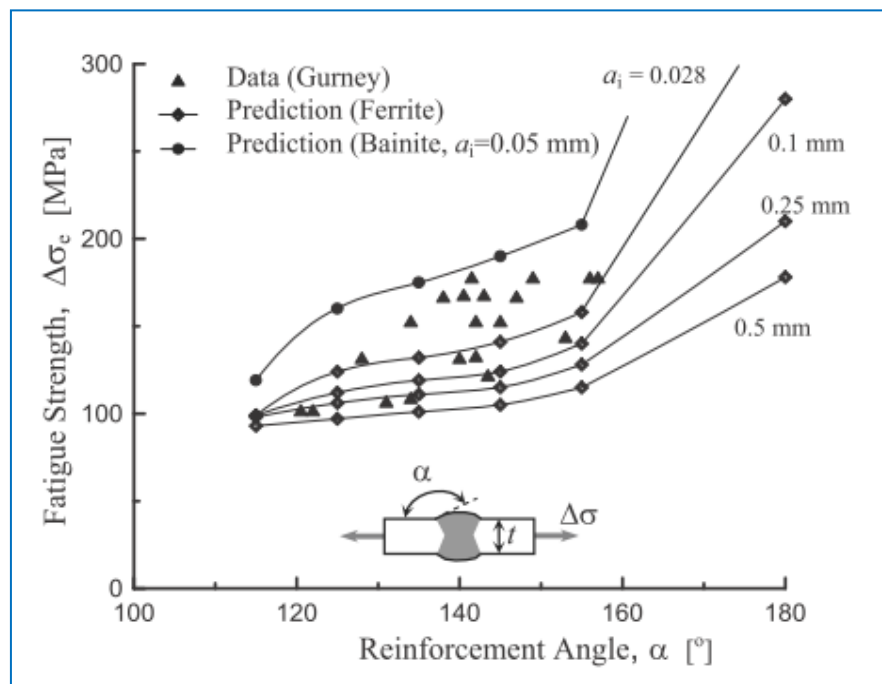


Figure 2-9: Reinforcement angle influence on fatigue strength of butt welds (Chapetti, 2011)

Pyttel, Grawenhof and Berger (2012) studied the influence of different welding geometries and welding procedures on the fatigue life of rotating components. Numerical and experimental investigations with different concepts such as nominal stress, hot spot stress, elastic notch stress and fracture mechanics were carried out. They found that most of the experimental results on the basis of nominal and hot spot stress concepts were conservative in comparison to the International Institute of Welding (IIW) recommendations and that the fatigue life of a laser-welded flat specimen was higher than that of the (tungsten inert gas) TIG weld specimen.

Radaj, Sonsino and Fricke (2009) review recent developments in the local approaches to the fatigue assessment of welded joints. A novel notch stress intensity factor (NSIF) and crack propagation approach for spot welds was reviewed. In this method the elastic stress intensity factor of the V-notch at the weld toe or at the weld root is determined and compared with durability values represented by a stress intensity – life curve.

2.9 Conclusion

- The geometrical parameters of welded joints such as weld toe radius, weld toe angle, weld reinforcement and weld width play a very important role in the fatigue strength of welded joints.
- Stress concentration generally occurs at the weld toe, which behaves as a pre-existing crack, and can be considered as a critical area for crack initiation with subsequent crack propagation leading to failure.
- All the research reviewed either investigates the effect of one or two geometrical parameters on the strength of welded joints or studies the welding parameters such as voltage, current and welding speed. However there is no comprehensive study taking in to account the effect of all geometrical parameters (weld toe radius, weld toe angle, weld reinforcement and plate thickness) on the strength of welded joints.

Chapter 3

Optimisation

Chapter Three: Optimisation

3.1 Introduction

One of the simplest definitions for optimization is “doing the most with the least” (Gomez, *et al.* p. 301, 2006). Lockhart and Johnson (1996) define optimization as “the process of finding the most effective or favorable value or condition” (p. 610). However, optimisation can be defined as the process of finding the conditions that achieve the maximum or minimum value of a function (Rao, 2009).

Optimisation problems are either scalar or vector optimisation problems. Scalar optimisation problems use a single objective function f and can mathematically be defined as “min/max $f(x)$ ”, whilst vector optimisation problems use several objective functions. The major categories of scalar optimisation are linear optimisation, quadratic optimisation, nonlinear optimisation, simplex method and quasi-Newton method. (Rao, 2009)

Moreover, the optimal seeking techniques are identified as mathematical programming techniques and are commonly considered as a component of operations research. Operations research is defined as the branch of mathematics which involves the application of scientific or systematic approaches and techniques to decision making problems as well as setting up the highest or optimal solutions.

Mathematical programming techniques are practical methods applied in order to obtain the lowest of a function of a number of variables governed by an arranged group of constraints.

3.2 Historical Development

The first emergence of optimisation methods dates back to the time of Newton, LaGrange, and Cauchy. It was possible to make the advancement of differential calculus methods of optimisation due to the contributions made by Newton and Leibnitz to calculus. Bernoulli, Euler, LaGrange and Weirstrass established the basics of variations calculus, which involves the reduction of functions. As for the method of optimisation for constrained problems, which is concerned with the addition of unknown multipliers, it is named after its inventor, LaGrange. Cauchy developed the earliest practice of the sharpest descent method in order to solve unrestricted minimisation problems. Regardless of these early contributions, no great advances were achieved until high-speed digital computers brought about the execution of optimisation

methods and motivated more updated research on new techniques in the middle of the twentieth century. Then enormous progress was made, resulting in an immense amount of work and publications on optimisation procedures. Furthermore, these advances led to the development of numerous distinctive new domains in the theory of optimisation (Carmichael, 1981).

It is noteworthy that the most important developments involved in the field of numerical methods of unconstrained optimisation were introduced into the United Kingdom only in the 1960s. Actually, the emergence of the simplex method devised by Dantzig in 1947 for linear programming problems and the annunciation of the principle of optimality in 1957 by Bellman for dynamic programming problems preceded the introduction of the methods of constrained optimisation by Kuhn and Tucker in 1951 on the adequate conditions for achieving the best solution of programming problems. Subsequently, they established the basis for a large amount of later research in nonlinear programming.

Zoutendijk (1960) contributed extensively to nonlinear programming in the period of the early 1960s. The work of Fiacco and McCormick (1968) made the solution of many difficult problems possible by employing the recognised techniques of unrestricted optimisation geometric programming which were developed by Duffin, Peterson and Zener (1967) despite the fact that no single technique has been proven to be uniquely appropriate for nonlinear programming problems.

Gomory (1963) conducted revolutionary work in integer programming, one of the most interesting and speedily improving fields of optimisation. This is due to the fact that most real-world uses are classified under this group of problems. Additionally, Charnes and Cooper (1959) introduced stochastic programming techniques and solved problems through acquiring plan constraints to be independent and typically distributed. The development of multi-objective programming methods was attained because of the desire to optimise more than one goal or purpose while fulfilling the substantial restrictions. Goal programming can be defined as the recognised method of solving definite types of multi-objective optimisation problems. It was initially planned for linear problems by Charnes and Cooper (1961).

Von Neumann set up the fundamentals of game theory in 1928 (Rao, 2009) and, from that time on, this procedure has been utilised to solve a number of mathematical economics as well as military problems. However, throughout the past few years game theory has been employed so as to solve engineering design problems. In fact, during the last decade simulated annealing, genetic

algorithms, as well as neural network techniques indicate a recent group of mathematical programming methods that have emerged.

Genetic algorithms are defined as search techniques made on the basis of the mechanics of evaluation and natural selection. Neural network methods are established on solving the problem by applying the proficient computing power of the unified "neuron" processors' network.

3.3 Optimisation Engineering Applications

Optimisation, broadly speaking, can be employed to solve any engineering problem. For the purpose of pointing out the broad scale of the area under discussion, some standard applications from the diverse engineering branches are listed as follows:

- Aerospace engineering: - optimisation of aircraft and aerospace configurations for minimum weight, aerofoil shape.
- Civil engineering structures: - e.g. frames, foundations, bridges, towers, chimneys, reinforced concrete structures and dams for least cost.
- Minimum-weight design of structures for earthquake
- Setting up water resources systems for optimal benefit.
- Structural engineering can be divided into three categories, i.e. sizing; shape; topology optimisation.
- Mechanical Engineering: - creating the most advantageous design of linkages, cams, gears, machine tools and other mechanical apparatus.
- Selecting corresponding settings in metal-cutting procedures for minimum production cost.
- Fabricating significant material handling machinery, for instance, conveyors, trucks and cranes for least economic cost.
- Chemical and process engineering: - planning pumps, turbines, heat exchanger and heat transfer equipment for maximum efficiency.
- Creating optimal electrical machinery, for example, motors, generators and transformers.
- Analysing statistical data and constructing experimental replicas from empirical findings to attain the most exact depiction of the material event.
- Microwave engineering: - integrated radar and communication system for wave form optimisation and optimisation of antenna.

3.4 Optimisation Problem

An optimisation problem may be represented as shown below (Moris, 1982):

$$\text{Find } X = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_n \end{Bmatrix} \quad \text{which minimise/maximise } f(x) \quad (3.1)$$

Subject to constraints:

$$g_j(X) \leq 0, \quad j = 1, 2, \dots, m$$

$$I_j(X) = 0, \quad j = 1, 2, \dots, p$$

Where X is an n -dimensional vector identified as the design vector, $f(x)$ is called the objective function and $g_j(x)$ and $I_j(x)$ are the inequality and equality constraints, respectively. Eq. (3.1) illustrates the problem called a constrained optimisation problem. Some optimisation problems are not concerned with any constraints and can be solved as:

$$\text{Find } X = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_n \end{Bmatrix} \quad \text{which minimise/maximise } f(X) \quad (3.2)$$

Such problems are called unconstrained optimisation problems.

3.4.1 Design variables

Any engineering system or constituent can be described according to a group of parameters. The first step in the formulation of an optimisation problem is to identify the underlying design variables. Each design problem usually contains many design parameters. The design's performance may be highly sensitive to some of these design parameters. These parameters may also be referred to as design variables.

Sensitivity analysis determines the importance of each input design variables on the system response. This is useful to identify which parameters can be excluded from the design

investigation to save computational time.

The first rule of thumb of optimisation problem formulation is to identify as few design variables as possible. The outcome of this initial optimisation investigation may indicate whether to replace previously considered design variables with new design variables or to include more design variables. In general, all design variables are restricted to lie within maximum and minimum limits as follows:

$$x_i^L \leq x_i \leq x_i^U \text{ for } i=1, 2, 3 \dots N \quad (3.3)$$

3.4.2 Design constraints

It is not possible in many practical problems to select design variables randomly; instead, they must fulfil clearly defined functions as well as other requirements. The limitations that must be fulfilled in order to develop a satisfactory design are commonly called design constraints. Constraints that signify restrictions on the desired behaviour or performance of the system are called criteria constraints. Constraints indicating physical limitations on the design variables are termed design variable constraints. The system may have additional constraints applied which are not related to the criteria but are functions of the design variables. These constraints are termed functional constraints.

For example, maximum stress will often be a constraint applied to the structure to ensure that it will not fail. However, minimizing or maximizing this stress may not be an objective of the optimisation. If a structure has uniform shape, the relation between maximum stress and dimensional design variables can be expressed in mathematical form. Otherwise, in the case of an irregular shape, a finite element simulation or other technique may be used to compute the maximum stress.

From most cases there are two types of constraints:

1. **Equality constraints** the constraints can divide into two types hard or soft, hard equality constraints are difficult to satisfy if the constraint is nonlinear in decision variables. Such hard equality constraint may be possible to relax (or made soft) by converting to them into an inequality constraint and adding some tolerances on them with some loss some of accuracy (Deb, 1995).

Example:

The temperature $T(\mathbf{x})$ of a point in the structure should be exactly equal to 10 C.

Then: $T(x) = 10$ in such cases it is difficult to deal with the equality constraints in the algorithms; equality constraint is changed to two inequality constraints and hence introduce a tolerance.

Example:

Previously $T(x) = 10$

Now it is changed to inequality constraints as shown below:

$$T(x) \geq 9$$

$$T(x) \leq 11$$

2. Inequality constraints state that the relationships between variables are greater than or smaller than a certain value.

Example:

The stress $\sigma(x)$ developed in a component during design must be smaller than or equal to the yield strength (σ_{yield}) of the material

$$\sigma(x) \leq \sigma_{\text{yield}}$$

Figure 3.1 illustrates a theoretical 2d design space where the non-feasible region is represented by hatched lines. A design point that is positioned on one or more constraint surfaces is termed a node point and its related constraint is referred to as an *active constraint*. Those points that are not placed on the constraint surface are called *free points*. According to the location of a design point on the design space, it can be grouped into four as follows:

- A free and acceptable point
- A free and unacceptable point
- A bound and acceptable point
- A bound and unacceptable point

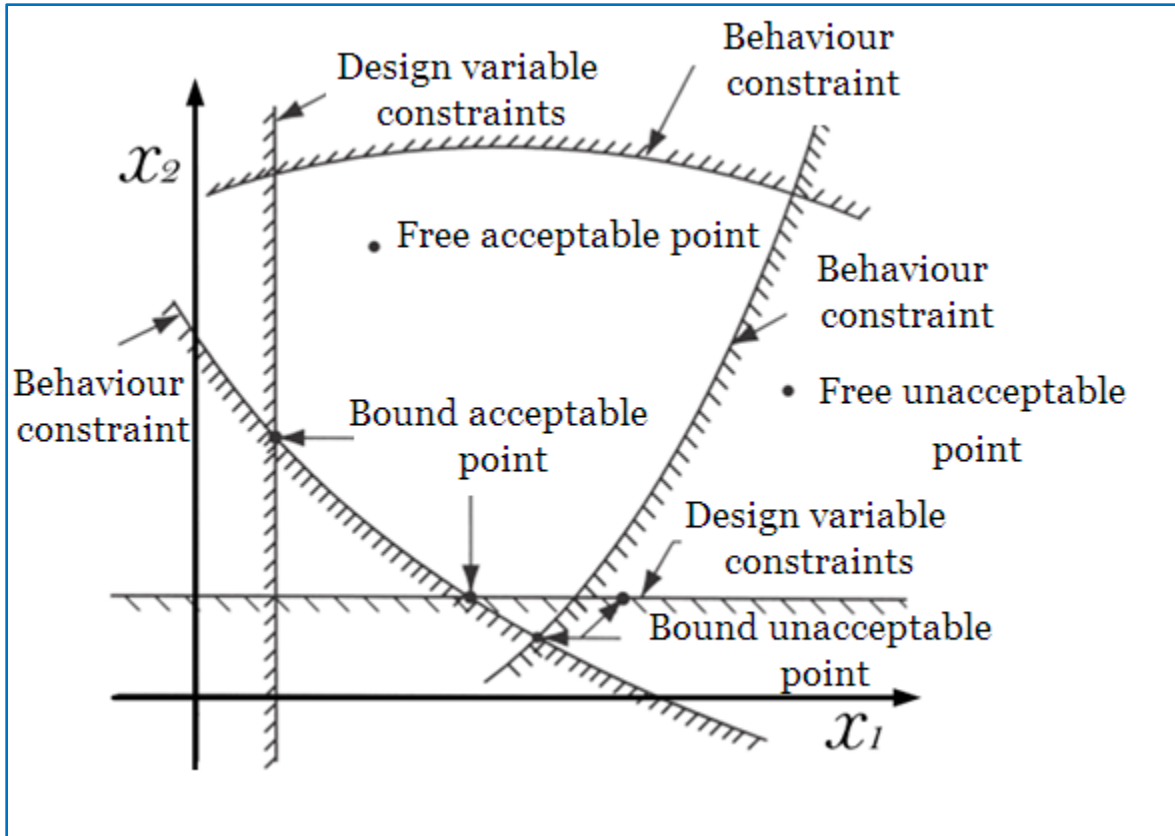


Figure 3-1: Constraint surfaces

3.4.3 Objective function

Achieving a satisfactory or efficient design which merely satisfies the functional and other requirements of the problem is the objective of the conventional design procedures. Broadly speaking, more than one acceptable design will be possible, and the target of optimisation is to select the best one of the several acceptable designs that is available. As a result, a standard should be selected for the purpose of drawing a comparison between the diverse alternatives in order to choose the best one. The standard, concerning which design is optimised, when represented as a function of the design variables, is identified as the criterion, merit or objective function. Likewise, the selection of the objective function is regulated by the nature of the problem. For example, the objective function for minimisation may be weight in aircraft and aerospace structural design problems. On the other hand, in civil engineering structural design, a commonly used objective is reduction of cost. In contrast, the maximisation of mechanical effectiveness may be the objective in mechanical engineering systems design. So, in aircraft design an alternative objective function could be to maximise the range or passenger volume and try to minimise the life cycle cost or specific fuel consumption.

Accordingly, the choice of the objective function can be simple or clear-cut in some design problems. However, there may be other situations where optimisation with regard to a specific criterion may bring about outcomes that may not be considered acceptable with reference to another criterion. For instance, in the field of mechanical design, a gearbox able to transmit maximum power may not possess the lowest weight. Equally, the minimum-weight design, in structural design, may not match the acceptable stress design. Accordingly, the choice of the objective function can be regarded as among the most important decisions in the whole optimum design process.

3.5 Population-based Optimisation Methods

This term is used to describe a search algorithm that is stochastic and population-based. This kind of algorithm is much more likely to yield a global optimum depending on the simulation of statistical mechanics.

3.5.1 Simulated annealing optimisation

A method was developed by Kirkpatrick, Gelatt and Vecchi (1983) as a way to simulate metals annealing. By annealing we mean the crystallisation of metals at a higher temperature through heating followed by slow cooling. Annealing derives from the fact that the atoms of metals carry higher energy and more speed at high temperature. Slow cooling provides us with the opportunity of keeping this energy and high rate in the movement of atoms for a longer time and avoids localised crystallisation. This allows the solution space to be fully explored. Quenching, on the other hand, tends to introduce localised crystallisation. This is analogous to local optimum solutions being found. Table (3.1) shows the main concepts of the simulated annealing method.

Table 3-1: Basic concepts of Simulated Annealing

Solid state	feasible solutions
Energy	fitness value
Minimum energy state	optimum value

Simulated annealing applies a high degree of temperature in order that the whole solution space is randomly searched. While it is being searched, the temperature goes down steadily and slowly toward the minimum state. The three elements that define simulated annealing are described as follows (Birk & Harries, 2003).

- **Annealing Schedule:** This is the schedule that determines how temperature is decreased over time.
- **Generating Probability-density Function:** Probability of generating new solutions depends on the random disturbance created about the current optimal solution. The generated probability density function forms the probability of disturbance. Greater disturbances are generated at high temperature. As temperature goes down the variance of the generated probability density function is reduced.
- **Acceptance Probability Density Function:** The outlined scheme determines the acceptance of new solutions. When the new solution $f(x_{(k+1)}) > f(x_{(k)})$, then $x_{(k+1)}$ takes the place of the older solution $x_{(k)}$. A random number $r_n \in (0,1)$ is created in the case when $E(x_{(k+1)}) < E(x_{(k)})$. In these cases the Boltzmann test is applied as follow:

$$e^{-\frac{E(x_{(k+1)})-E(x_{(k)})}{T_{obj}}} \begin{cases} > u \text{ the new design } x_{(k+1)} \text{ is accepted} \\ \leq u \text{ the new design } x_{(k+1)} \text{ is rejected} \end{cases} \quad (3.4)$$

3.5.2 Particle swarm optimisation

This is a type of optimisation method that takes its basic concept from the behaviour of large groups of social animals. This may be a swarm of bees looking for a hive location or a flock of birds looking for food or a place to roost. It is a stochastic and population-based method. Eberhart and Kennedy (1995) were the first to check the validity of this method in optimisation. It is found that many problems of optimisation, as in Genetic Algorithms, can be worked out through the Particle Swan Optimisation technique.

The PSO system depends on the creation of a number of particles regarded as a swarm that aim at checking and flying over the hyper-dimensional solution space simultaneously. The mission of each single particle is to record their personal best position and read both local best (lbest) and swarm's best position (gbest). The velocity vector in such types of search is a driving factor that directs the particles positions so that they can be improved. An inertia factor, W , determines the influence that the previous velocities have on the current one. Additional factors

such as cognitive and social factors are brought in to control the particle's confidence in itself or in the swarm. The main duty of the cognitive factor, $C1$, is to determine the level of confidence in success for each particle. The social factor $C2$ is responsible for detecting that confidence level. Table 3.2 shows the standard PSO nomenclature.

Table 3-2: Nomenclature of Particle Swarm Optimisation

Symbol	Meaning
p_i	Particle i
$x_i(t)$	Particle position at time t
$vi(t)$	Particle velocity at time t
I_{best}	Best position found by particle
g_{best}	Best position found by swarm
W	Inertia factor
$C1$	Cognitive factor
$C2$	Social factor

3.5.3 Genetic Algorithm

Genetic Algorithms (GA) are optimisation methods that are non-deterministic and population-based. It was Holland, (1975) who brought this technique to light. What marks this method is its dependence on imitating natural evolution: only the fittest will survive. In other words, the genetic properties of the parents are changed so that new generation of individuals will be fitter than the previous ones. For this change, mutation and crossover are used among other genetic processes to achieve the desired effect. Of course, the global optimum is the utmost objective of these genetic operations, so they are modified and set for this purpose.

Figure (3.2) shows the optimisation process of a GA:

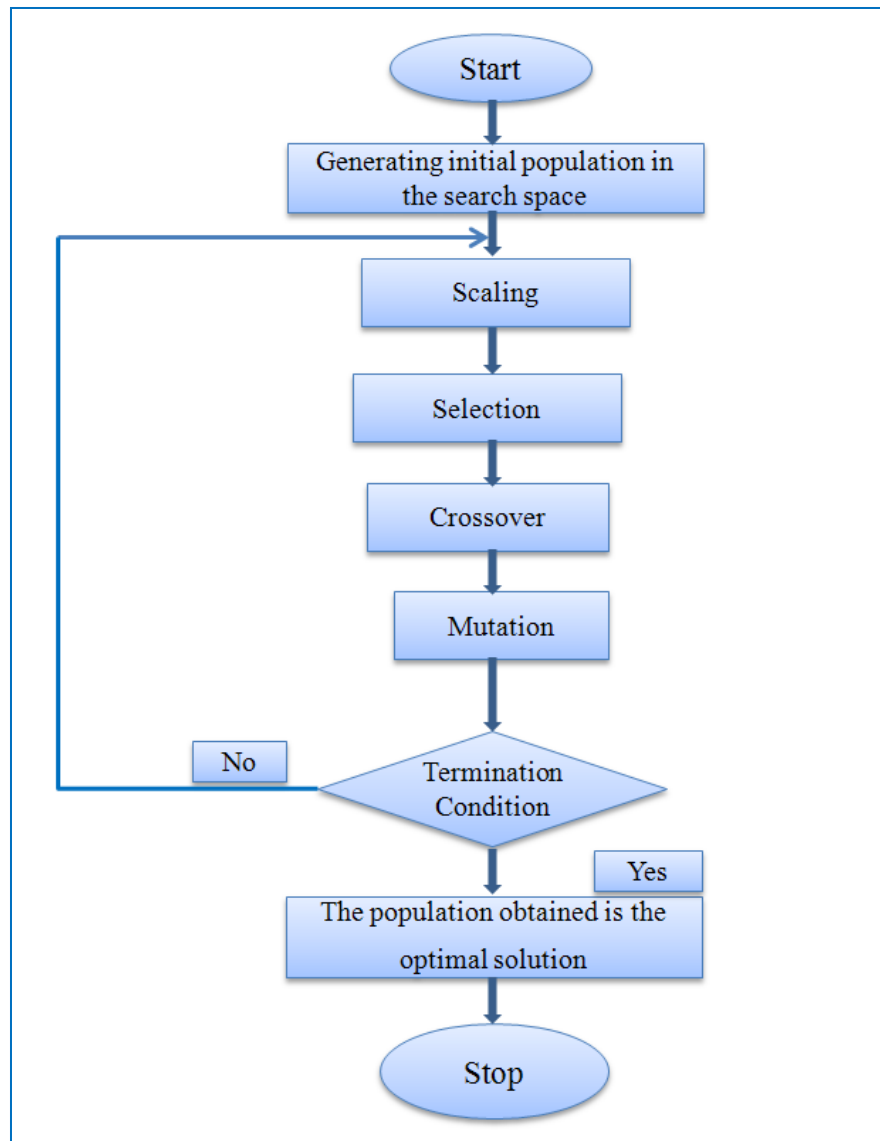


Figure 3-2: Genetic algorithm flow chart

Currently, there are four major genetic operations that create the cornerstone of the genetic algorithm technique:

1. *Tournament Selection*: Through this operation the fittest element from a population is selected after random sampling.
2. *Crossover*: This is to randomly change the positions of one or more bits located between two strings. There are several crossover operators.
 - *Single-point crossover*: This is the operation whereby children are generated through swapping data bits (chromosomes) that are located beyond one crossover point on a parent chromosome string. The swapped chromosomes are between two parent

chromosomes. Two children are born from each mating process.

Chromosome 1	11011 00100110110
Chromosome 1	11011 11000011110
Offspring 1	11011 11000011110
Offspring 2	11011 00100110110

- *Multiple-point crossover*: Unlike the single-point crossover, where one crossover point is selected on each parent, the multi-point crossover means that a number of crossover points are selected on every parent string. Between these selected points, all information is swapped.

Chromosome 1	11011 00100 110110
Chromosome 1	10101 11000 011110
Offspring 1	11011 11000 110110
Offspring 2	10101 00100 011110

- *Uniform Crossover*: in this operation, previous information is used to suggest the selection of crossover points based on a probability factor, so the operation is considered a biased one.

Chromosome 1	11011 00100 110110
Chromosome 1	10101 11000 011110
Offspring	10111 00000 110110

3. *Mutation*: this is to modify bits at random positions to maintain diversity within the population and inhibit premature convergence.
4. *Elitism*: The fittest elements form a generation survive through the next generation.

In order to reduce the computation burdens whilst improving the search process, a number of genetic operations and selection methods for GA's have been examined for many. The most popular of these will be investigated in Chapter 4.

3.6 Multi-objective Optimisation

Unlike uni-criterion optimisation, which requires only one objective function, complex engineering optimisation has its own problems with a number of different criteria having to be satisfied at the same time.

3.6.1 Problem definition

Multi-objective optimisation demands simultaneous optimisation of several objectives based on a collection of both decision variables and constraints. The problem for such optimisation can be summarised in the following words: “Find the optimum of objective functions when the decision variables are subject to inequality and equality constraints”.

Mathematically, the problem of multi-objective optimisation is seen in the following formula: Find the set of optimal solutions ($f(x^*)$) with respect to a number objective functions ($f(x)$).

The basic elements of this problem are listed below:

Decision Variable Vector for n variables

$$x = [x_1, x_2, \dots, x_n]^T \text{ for n variables} \quad (3.5)$$

Equality Constraints

$$h_j(x) = 0 \quad \text{where, } j=1, \dots, p \quad (3.6)$$

Inequality Constraints

$$g_j(x) \geq 0 \quad \text{where, } j=1, \dots, m \quad (3.7)$$

Objective Function Vector for k objectives

$$f(x) = [f_1(x), f_2(x), \dots, f_k(x)]^T \quad (3.8)$$

The following sections will discuss and analyse some different schemes that have been created to deal with multiple-objective functions.

3.6.2 Handling Multi-objectives

Optimal solutions are the main target that is sought in any multi-objective optimisation. Therefore, a number of schemes have been formed to produce this sort of solution. The

following three approaches are viewed as the major ones developed for this aim.

Aggregative approaches: The advantage of these approaches is that one solution is gained following every single run. The approaches gather all the objectives in one function, which is a high scalar function, in a linear way. This function is employed in the fitness calculation. Awareness of the solution space is a prerequisite to determining the weights to apply to the individual objective functions. This is a drawback of these methods. Goal attainment, target vector optimisation and weighted sum methods are popular aggregation techniques.

Population-based non-Pareto approaches: In this kind of technique, the selection criterion is changed while the search is conducted in order to allow for concurrent multiple-direction searches. Working in parallel, the objective vectors are estimated and added, but there is no guarantee that an approximation to the optimal solutions set can be reached.

Pareto-based approaches: In the Pareto-based approaches, the optimal solutions set is mapped using Pareto dominance. In the following section, this process is illustrated.

3.6.3 Pareto dominance

Historically, the Pareto dominance method was developed by Vilfredo Pareto, an economist of Italian and French origin, at the beginning of the twentieth century. The Pareto-optimal (x^*) is considered a solution when there are no other reasonable solutions (x_i) that could reduce a certain criterion while not resulting in the concurrent increase of any of the other criteria. The mathematical expression of Pareto optimality minimisation can be viewed as follows:

The Pareto-optimal solution $x^* \in X$ appears when for every $x \in X$ and $I=(1,2,...,n)$ either

$$f_I(x_i^*) < f_I(x_j) \wedge f_2(x_i^*) < f_2(x_j) \wedge \wedge f_k(x_i^*) < f_k(x_j) \quad (3.9)$$

Or at least there is one $f_h(x)$ such that

$$f_h(x_i^*) < f_h(x_j) \text{ and } f_h(x_i^*) \neq f_h(x_j) \quad (3.10)$$

Pareto-optimal solutions are also called non-dominated, non-inferior, or efficient solutions.

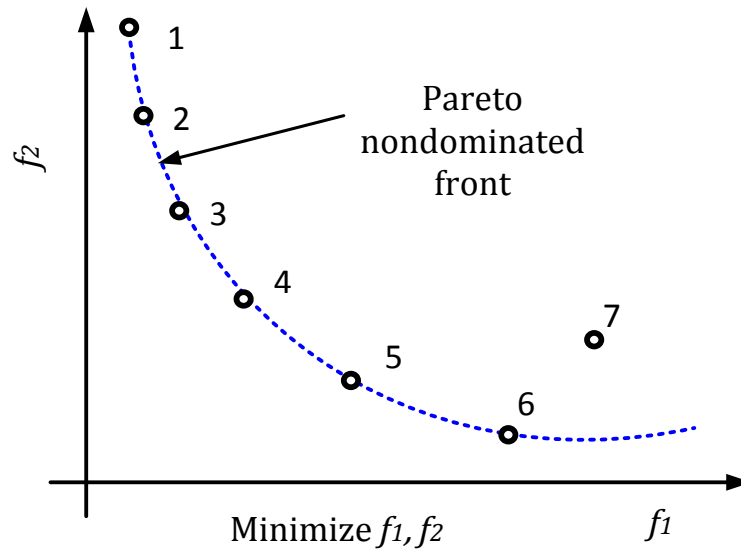


Figure 3-3: Main concept of Pareto dominance in a two objective problem

In Figure 3.3, solution 7 is dominated by solution 6, as 6 is better than 7 in both objectives. Solution 1 does not dominate solutions 2, 3, 4 and 5. In this case 1, 2, 3, 4, 5 and 6 form a non-dominated or Pareto front solution.

3.6.4 Classical methods

According to various researchers and scholars there are two techniques for solving multi-objective functions: (1) classical methods; (2) evolutionary methods. According to Shukla, Deb and Tiwari (2004), the classical methods of multi-objective optimisation utilise gradient or direct-based techniques founded on various mathematical principles. In addition, the three scholars identify a number of classical multi-objective methods: the epsilon-constraint technique, the weighted sum, Schaffler's stochastic method, Timmel's population-based method and the normal boundary intersection method.

Another scholar and researcher who examines and explores the classical methods of multi-objective optimisation is Zitzler (1999). He studies and examines two classical methods of multi-objective optimisation, namely: the weighting and constraint methods. He concludes that the weighting method may be sensitive to the shape of the Pareto-optimal front.

Research was conducted by Sbalzarini *et al.* (2009) to investigate and examine various tools that can be utilised for optimisation of the welding processes. Sbalzarini *et al.* (2009) discovered that individuals, institutions and organisations resorted to using genetic algorithms and evolution

algorithms because of their ability to solve complex problems. They further noted that earlier optimisation techniques, which included gradient and linear programming methods and models, could not solve complex optimisation problems and challenges.

Chen and Banet, (2010) further cite the limitations of utilising linear programming models in the optimisation of welding processes. In addition, the two scholars note that the linear programming model does not produce an optimal solution in a timely manner.

3.6.5 Evolutionary multi-objective optimisation

Over the past 20 years, Evolutionary Algorithms (EAs) have been widely utilised for Multi-objectives Optimisation problems since they use a strategy based on a population of solutions. The conflicts resulting from the multiple objectives have stimulated the development of EA techniques that avoid weighting (compromising) the different objectives. One of the significant advantages of the EA method lies in the fact that solutions in the Pareto front are distributed in a well-spread way.

The problems and difficulties encountering in using multi-objective evolutionary algorithms (MOEAs) have been discussed widely in the literature. A clear view on the common and widespread difficulties of MOEAs can be found in Zitzler, Deb and Thiele's (2000) Technical Reports I and II. However, much research into multiple-objectives optimisation has supported the concept of elitism adopted by EAs. This concept, which depends on the idea of 'survival of the fittest' for the best solutions, has been experimentally and theoretically validated either directly or indirectly by many different researchers and implementations, such as the Non-dominated Sorting Genetic Algorithm (NSGA-II) by Deb *et al.* (2002), the Pareto Envelope-based Selection Algorithm (PESA) by Corne *et al.* (2000), the Pareto Archived Evolution Strategy (PAES) by Knowles and Corne (1998) and the Strength Pareto Evolutionary Algorithm (SPEA and SPEA2) by Zitzler and Thiele (1998 and 1999).

Basically, there are four main categories under which evolutionary algorithms fall.

- 1- Evolutionary Strategies (ES).
- 2- Evolutionary programming (EP).
- 3- Genetic Algorithms (GA).
- 4- Genetic Programming (GP).

EAs exhibit an adaptive behaviour that allows them to handle high dimensional, nonlinear problems without requiring explicit knowledge of the problem structure or differentiability.

The multiple categories refer to the different specialisations where this technique can be employed, though the principles found in for each category are relatively different from one another. More detailed information about evolutionary algorithms and comparisons to traditional multi-objectives optimisation can be found in many authors research, such as Branke *et al.* (2008a), Poli *et al.* (2008), De Jong (2006), and Deb, (2001).

In recent years, different EAs have been adjusted in numerous approaches to handle multi-objective Pareto optimization. Among many algorithms, one of the EAs for multi-objective optimization is MOGA-II described by Poles (2003), MOGA-II is an improved version of MOGA (Multi-Objective Genetic Algorithm) by Poloni (1997), and bearing in mind that MOGA is not the same as Fonseca and Fleming's MOGA. Fonseca and Fleming introduced a niching scheme that calculates distances in the criteria space (in contrast to distance measurement in decision variable space) Fonseca (1993), MOGA-II uses a smart multisearch elitism for robustness and directional crossover for fast convergence. Its efficiency is ruled by its operators (directional crossover, classical crossover, mutation and selection) and by the use of elitism.

Poles (2004) presented and tested MOGA-II on single-objective optimization problems (with and without noise). They compared it to differential evolution and a standard evolutionary algorithm, they concluded that the MOGA-II sometimes performed better and never worse than both algorithms.

Farnsworth *et al* (2011) compared the performance of the most referenced MOEAs (NSGA-II and MOGA-II) in the design optimisation of micro electromechanical systems (MEMS) through a number of case studies of increasing complexity. They provided extended discussion and analysis of the results which showed, overall that MOGA-II outperformed NSGA-II, for the case studies.

The comparison of performance between NBI-NLPQLP a multi-objective scheduler based on the Normal-Boundary Intersection (NBI) and MOGA-II have been applied to the five benchmark problems (No-hole, DEB constraint, TNK constraint, POL and Hole problem) by Rigoni and Poles (2005) indicated that MOGA-II outperformed NBI-NLPQLP with high robustness.

A Fast Multi Objective Genetic Algorithm (FMOGA-II) based on modified version of MOGA-II was presented by Rigoni (2010). The fast prefix refers to the rate of convergence - in terms of iterations needed - towards the solution of the problem. This fast optimizer used metamodeling to speed up the process: During the virtual optimization process the FMOGA-II is run over the best available meta-models. The population for the next iteration the database is built up of 50 % of the points from the current Pareto set and 50 % random points from a surface approximating the Pareto surface. The Pareto front points contribute to faster convergence and the random points increases the robustness of the optimizer.

A multi-objective shape optimization study of a tube bundle in heat exchanger has been conducted by D. P. Ranut (2012), he tested two different Genetic Algorithms NSGA-II and FMOGA-II, and the results confirmed that FMOGA-II is a promising algorithm for reducing the computational time in this type and similar optimisation problems.

The performance of FMOGA-II with respect to its direct precursor MOGA-II was tested by Rigoni and Turco (2010). The comparison focused on the ratio between the quality of the best-so-far solution (point or set) and number of evaluated designs. Two single-objective and two multi-objective problems were used during this study. The tests showed the possible advantages of fast optimizers.

3.7 Robust Optimisation

3.7.1 Variability in reality

There are different types of parameter that can influence the response or quality characteristics of a process. These parameters can be categorised into the following three types and presented in a block diagram of the process (Figure 3.4).

- 1- *Signal Factors*: These are the parameters specified by the operator or user to express the intended value for the response of the product. For example, steering wheel angle is used to specify the turning radius of a car and the speed setting of a fan is set to determine the velocity of an air flow.
- 2- *Noise Factors*: Parameters whose settings are difficult to control in the field or parameters which cannot be controlled by the designer or whose levels are expensive to

control. The levels of noise factors change from time to time, unit to unit and from one environment to another. These factors cause the response to deviate from the target.

- 3- *Control Factors*: Parameters which can be set freely by the designer. In order to obtain the least sensitivity of the response to the effect of the noise factor, the designer has to determine the best values for these parameters.

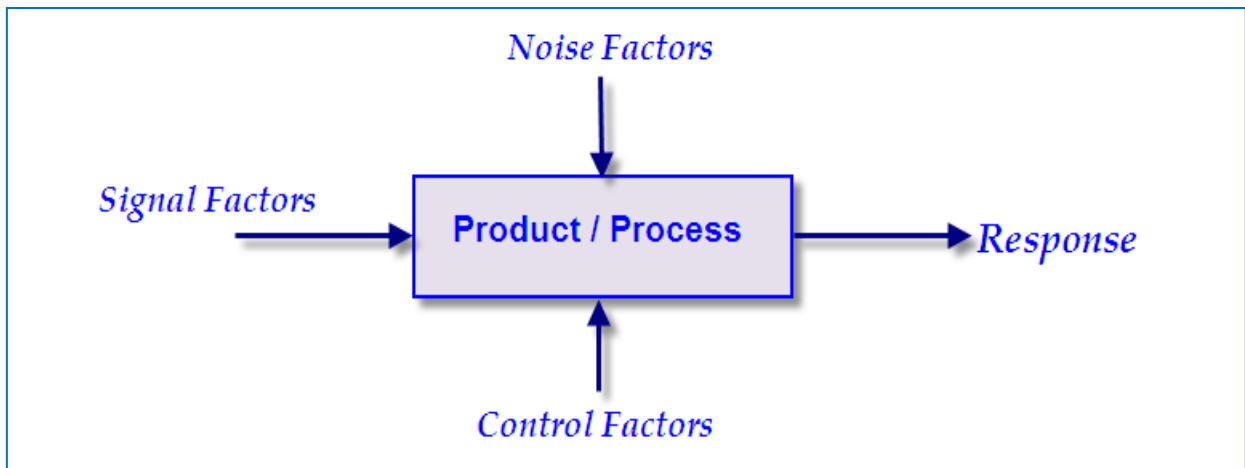


Figure 3-4: Block diagram of the Product/Process (Bagchi, 1993).

3.7.2 Robust design

Often engineering design cases with multiple objectives have design factors which vary uncontrollably because of uncertainties or noise. The results can be significantly influenced by such variations, to the extent that it may not be possible to achieve certain objectives and/or the ideal Pareto solutions may be impracticable. A robust optimisation approach with respect to the objective functions is desired to give the least degree of sensitivity to variations in the design variables and parameters. Practically speaking, no engineering design exhibits absolute insensitivity towards uncertainties evolved from manufacturing processes, modifications in material characteristics, operation set-up, etc. In addition, non-robust designs are costly in terms of manufacturing and show poor performance during service.

The solution of a robust optimisation problem with a single objective is demonstrated in Figure 3.5. When the design variable x is equal to x_{opt} , the performance function $f(x)$ is minimum. However, in x_{opt} , $f(x)$ shows considerable sensitivity towards variations. In fact, Δf_{opt} , which

represents the range of variations in $f(x)$ for a specified range of variations in x around x_{opt} , is large. On the other hand, x_{rob} is a local minimum of function $f(x)$ and the $f(x)$ shows negligible sensitivity when x_{rob} is varied.

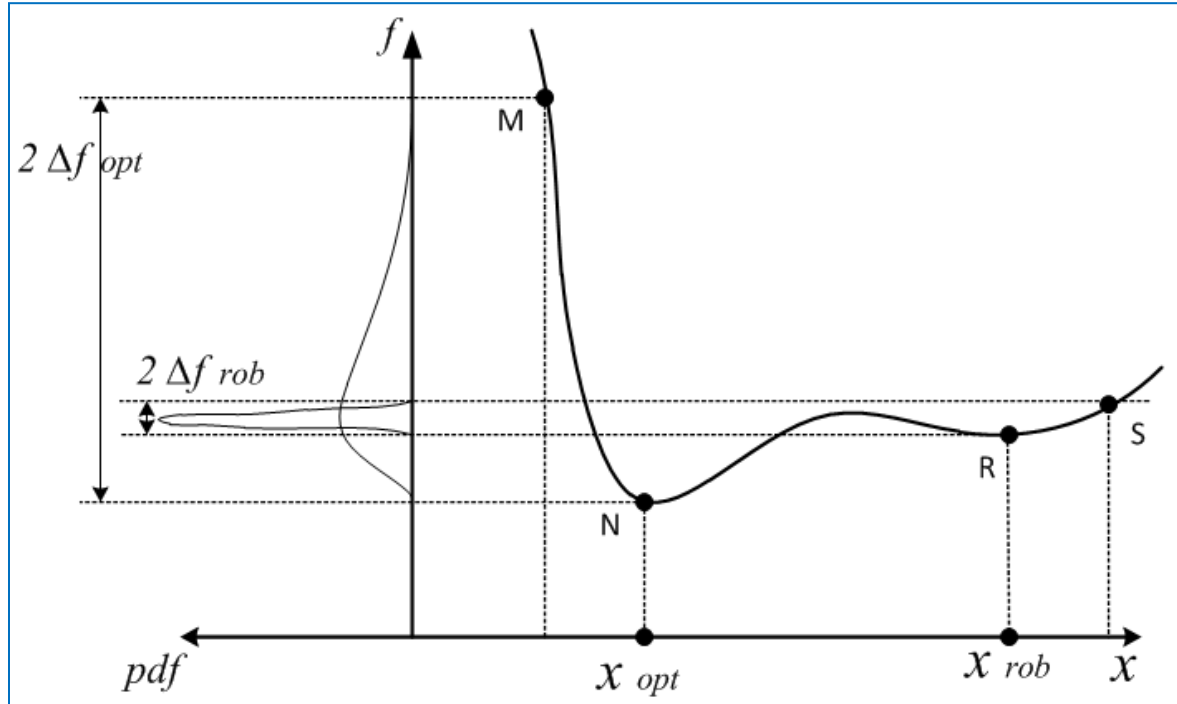


Figure 3-5: A robust solution and optimal solution

3.7.3 Robust multi-criteria optimisation

Taguchi (1993) was the first to introduce the notion of a robust design. He proposed the idea of a parameter based design to enhance the performance of manufactured items whose production entailed considerable inconsistency or noise. The purpose of robust design is to reduce the effects of variations on the performance of the product without eliminating the sources of these discrepancies. The efforts of the researchers have thus led to various developments and enhancements in robust design problems over the past several years.

A set of design parameters $\mathbf{p} = [p_1, p_2, p_3 \dots p_q]^T$ must be taken into account to achieve robustness. These parameters are out-of-control, i.e. beyond the power of the designer. For example, the price of the steel used to manufacture ships or production variations, wear on any other uncertainty, even though their nominal value remains fixed. The design variables are given as:

$$\mathbf{X} = [x_1, x_2, x_3 \dots x_n]^T$$

The goal of a typical multi-objective robust design optimisation problem is to discover the design variables which can optimise a vector objective function, $\mathbf{f}(\mathbf{X}, \mathbf{p})$. whilst endeavouring to reduce the range of variations it yields $\Delta \mathbf{f}(\mathbf{X}, \mathbf{p}) = [\Delta f_1, \Delta f_2, \Delta f_3, \dots \Delta f_k]^T$ over the practical design space S. Defining $\mathbf{v}^T = [\mathbf{X}^T \mathbf{p}^T]$, the problem can be stated as

$$\text{Minimise: } \mathbf{f}(\mathbf{X}, \mathbf{p}), \quad (3.11a)$$

$$\Delta \mathbf{f}(\mathbf{X}, \mathbf{p})$$

$$\text{Over } \mathbf{X} = [x_1, x_2, x_3, x_n]^T \quad (3.11b)$$

Subject to:

$$g_i(\mathbf{X}, \mathbf{p}) + \Delta g_i(\mathbf{X}, \mathbf{p}) \leq 0, \quad i = 1, 2, \dots m. \quad (3.11c)$$

$$\mathbf{X}_{\inf} \leq \mathbf{X} \leq \mathbf{X}_{\sup} \quad (3.11d)$$

$$\mathbf{v} - \Delta \mathbf{v}_{\inf} \leq \mathbf{v} \leq \mathbf{v} + \Delta \mathbf{v}_{\sup} \quad (3.11e)$$

Where the \mathbf{X}_{\inf} and \mathbf{X}_{\sup} are respectively the lower and upper bounds of the design variables, $g_i(\mathbf{X})$ represents the i^{th} inequality constraint function.

3.8 Conclusions

Genetic algorithms are well established as an optimisation tool and have been widely used for multi-criteria problems. There are a number of factors which must be taken into consideration when determining an optimisation algorithm; there are typically many standard parameters which can affect the performance of the optimisation, variable specification (probabilistic or deterministic), tight variable bounds and constraints. Therefore It is difficult to generalise on which is the “best” algorithm to use for all types of optimisation problems.

The concept of Pareto optimality is now well established within the field of multi-criteria optimisation and is generally accepted as being superior to the weighted criteria method.

A number of genetic algorithms have been developed which are able to accommodate multiple criteria without resorting to weighting methods. These various algorithms have individual merits in terms of the speed at which they can arrive at an approximation of the Pareto front (i.e. the size and number of populations required), the coverage of both the design space and the criteria space and their ability to avoid local minima. However, there does not appear to have been an independent study of the algorithms to compare their merits. All the comparisons found have been carried out by the developers of particular algorithms using particular case studies.

Robust optimisation is a technique that has developed to help ensure that products are not sensitive to manufacturing variations. This technique has been applied to multi-criteria problems to ensure that variability in all criteria is minimised. However, there appears to have been no attempt to ensure that the robust design remains close to the nominal parameter Pareto set.

Chapter 4

Optimisation Study

Chapter Four: Optimisation Study

4.1 Introduction

Instead of a single optimal solution, in multi-objective problems it is not possible to have a single solution that optimises all objectives. There usually exists a set of non-dominated solutions or Pareto optimal solutions. The mathematical formulation of multi-objective optimisation is discussed in detail in Section 3.6.1.

In the previous chapter it was concluded that whilst various multi-criteria genetic algorithms had been developed, no independent comparison of the merits of these algorithms could be found. The main objective of this chapter therefore, was to perform a comprehensive comparison between multi-objective optimisation methods on both engineering and mathematical problems, to determine the efficiency of each method and to determine if any particular algorithm performs well over a range of problem types when assessed using a number of criteria. The problems chosen from the literature included two simple mathematical problems with concave and convex solution sets in criteria space and a more demanding practical problem with interacting design variable, functional and criteria constraints and a large Pareto set surrounded by a feasible region in the design space.

Two of the measures to be used to assess the various multi-criteria genetic algorithms are the breadth and uniformity of the distribution of solution points on the Pareto front. The techniques available in the literature for quantifying these measures and their applicability to genetic algorithm generated solutions are therefore assessed in this chapter.

Six multi-objective population-based optimisation algorithms are introduced to compare the effectiveness of each. The six algorithms examined in this study are namely: Multi-objective Genetic Algorithm (MOGA-II) Poles (2003), Adaptive range Multi-objective Genetic Algorithm (ARMOGA) Daisuke (2005), Fast Multi-objective Genetic Algorithm (FMOGA-II), Non-dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2000), Multi-objective Particle Swarm Optimisation (MOPSO) (Sanaz, 2004) and Multi-objective Simulated Annealing (MOSA) Suppapinarm S, & Prka, (2000).

The problems which have been investigated are the following:

1. Optimisation of a lifting arm problem originally presented by Ghurbal (2003).

2. Multi-objective optimisation convex problem (SCH) suggested by Schaffer (1987).
3. Multi-objective optimisation concave problem (FON) used by Fonseca and Fleming (1998).

Different metrics implemented during the comparison study:

1. The variance of solution distribution in the design space regions (Coverage metric – COV), defined by Gunzburger and Burkdart (2004).
2. The ratio between the number of resulting Pareto front members to the total number of fitness function calculations (i.e. the Hit-Rate described by Sedenka & Raida, 2010)).
3. Circumscription metric (CM), defined by Tahernezahdiani *et al.* (2012).
4. Graphical representation of the Pareto fronts for qualitative assessment.

In the optimisation of a lifting arm Coverage, Hit-Rate metrics and one visual criterion were chosen for quantitative and qualitative comparisons.

These metrics were chosen to represent the quality as well as speed of the algorithms by ensuring good coverage of the solution spaces.

In the second (SCH problem) and third (FON problem) cases the comparisons were done by using two standard test problems which represent convex and concave optimisation problems. Another metric was used during this study, the Circumscription metric (CM) which indicates the distribution of the Pareto front in objective space in addition to the second metric from the previous study.

4.2 modeFRONTIER Software

The tool used to undertake the comparison of optimisation algorithms described above was modeFRONTIER (2008). This package is a multidisciplinary, multi-objective design optimisation code, written to allow easy coupling to different commercial computer-aided engineering (CAE) tools.

In this project modeFRONTIER was linked to Microsoft Excel where a closed form solution was possible and to the ABAQUS finite element code where a numerical model was necessary.

As explained later in the thesis, in order to achieve the objective of a robust and Pareto optimal solution, both these calculation methods were linked to modeFRONTIER simultaneously.

In general, to understand the modeFRONTIER, Figure 4.1 may be inspected, which indicates a simple example of both process design and optimisation.

With modeFRONTIER three main steps are essential for defining the optimisation problem:

- Parameterise the problem as a set of design variables that can be used to define a calculation or analysis to be carried out by the application package (Excel or ABAQUS in this case).
- Set objectives which define the performance of the design. It must be possible to extract these objectives as numerical values from the output file of the application package.
- Choose the strategy for optimisation in terms of the method used to generate the initial population and the algorithm to be used to determine the optimal population.

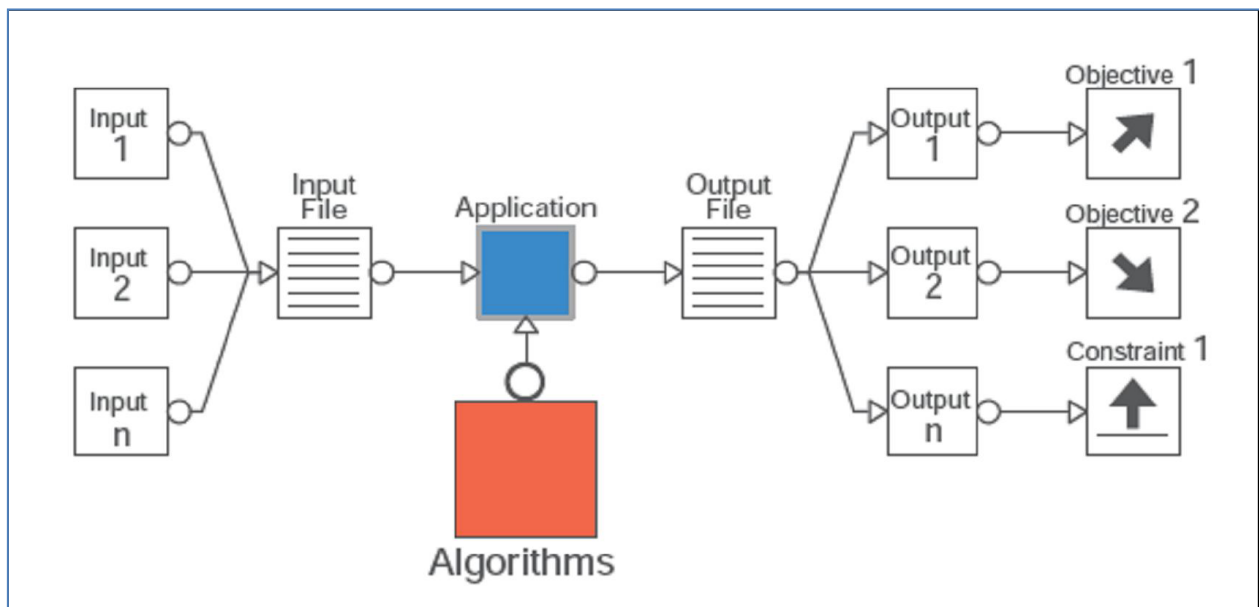


Figure 4-1: General modeFRONTIER process of integration and optimisation
modeFRONTIER document (2008)

4.3 Initial Population Generation

4.3.1 Design of Experiment (DoE) methods

As explained in the previous section, in order to initiate the optimisation process an initial population has to be defined. In order to do this a number of different design space search methods taken from DoE methodology can be selected. The objective in selecting a particular method is to search the design space thoroughly with the smallest number of points possible. So for example, whilst a full factorial search will be very thorough, it can generate a very large number of trial points where there are multiple design variables. Since the weld geometry analysis will use 6 design variables, a full factorial search of the design space would be unreasonable. Hence, a Sobol sequence (explained in section 4.3.3) has been used to define the initial population for each optimization analysis.

In order to implement the Sobol sequence, variables are normalised (scaled) and $-1 \leq x_i \leq 1$ is used as an interval. The u_1 and u_2 mapping can be Stromberg (2010).

$$u_i = \frac{x_i - \left(\frac{x_i^{max} + x_i^{min}}{2}\right)}{\left(\frac{x_i^{max} - x_i^{min}}{2}\right)} \quad (4-1)$$

When $x_i^{min} \leq x_i \leq x_i^{max}$ and ($i=1, 2$) are considered. Figure 4.2 shows the mapping process.

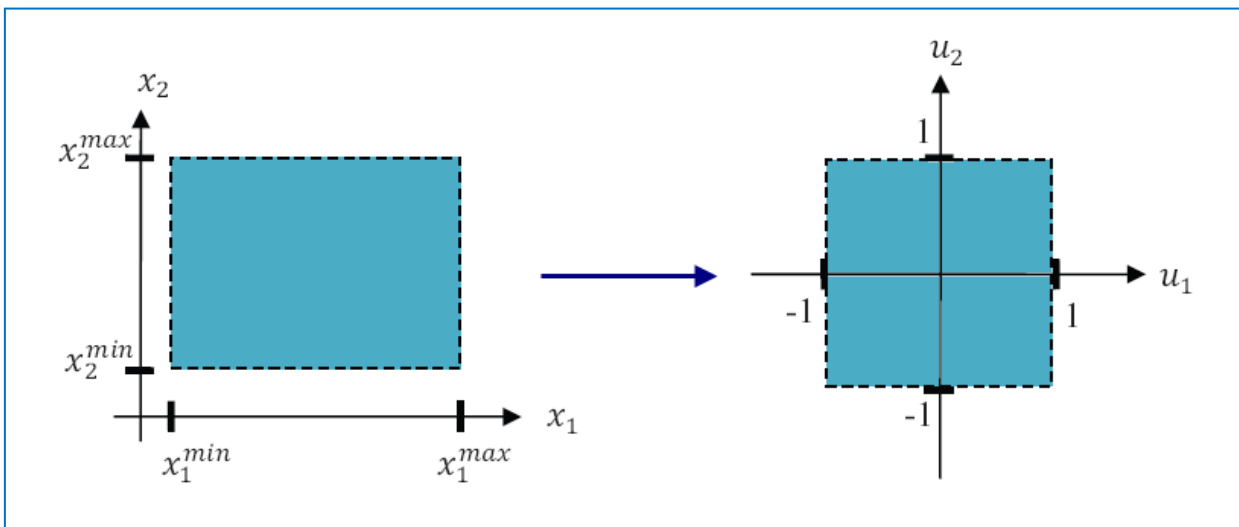


Figure 4-2: Normalisation and mapping of design variables

4.3.2 Measuring discrepancy

In a typical optimisation process, there is a need to generate a number of points and scatter them within a design space of a certain number of dimensions. How well-distributed these points are can be measured by considering the ‘discrepancy’ which can be defined in the following equations:

Definition

Let us consider a sequence of points $P_1, P_2, \dots, P_i, \dots$ belonging to a unit r - dimensional cube K^r . By G we denote an arbitrary domain in K^r , and by $S_N(G)$ the number of points P_i belonging to $G(l \leq i \leq N)$. A sequence (P_i) is called uniformly distributed in K^r , if

$$\lim_{N \rightarrow \infty} \frac{S_N(G)}{N} = V_G \quad (4.2)$$

Where $V(G)$ is the volume of the r -dimensional domain G

Source: Statinkove and Matsov, 2012

The meaning of the definition is that for a large value of N , the number of points for a given sequence belonging to an arbitrary domain G within the unit cube is proportional to the volume $V(G)$, i.e.

$$S_N(G) \approx NV(G) \quad (4.3)$$

4.3.3 Sobol Design

The Sobol design is alternatively termed pseudo-random or quasi-random. It is a sequence which gives a low discrepancy and hence a better distribution of points in the design space (Bratley & Fox, 1988).

A clear picture of the difference between Sobol and random sequence distribution is shown in Figure 4.3.

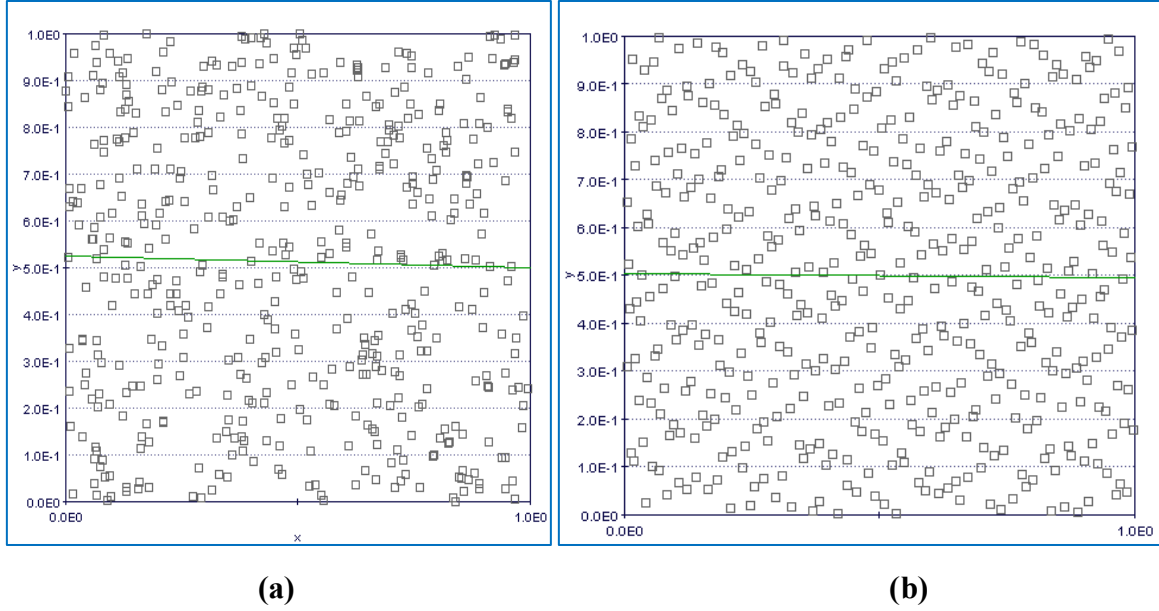


Figure 4-3: Distribution of the scatter plot of (a) pseudo-random sequence and (b) pseudo-random Sobol sequence in modeFRONTIER.

4.4 Measurement of Algorithm Performance

As mentioned before, the aim of this study was to compare multi-objective optimisation algorithms in terms of their performance on different test problems. Unlike single objective optimisation, in multi-objective optimisation the two aims are to determine the solution as close to the Pareto optimal as possible and to find a solution which is as diverse as possible in design space and objective space.

Therefore, there is necessary to have at least two performance metrics for evaluating multi-objective optimisation algorithms. The description of the metrics used is given below.

4.4.1 Hit-Rate Metric (HR %)

Different classifiers are used to describe the results. The number of trial points on the Pareto front is given by the parameters PF, while the parameters FFC denote the total number of trial points. The final hit rate HR is computed according to

$$HR = \frac{PF}{FFC} 100[\%] \quad (4-4)$$

A higher hit rate indicates that less calculation time was consumed to find Pareto optimal solutions. Hit rate can be used directly to compare different solution algorithms.

4.4.2 Pair Wise Metric (PW)

Other criteria are used to measure the uniformity of the distribution of points within the design space or criteria space.

In this study, the pair-wise metric defined by Gunzburer and Burkardt (2004) has been used in design space, which gives an indication of the diversity of points in design space. For a set of N points $\{z_i\}_{i=1}^N$ the minimum distance between point z_i and any other points is

$$\gamma_i = \min_{j \neq i} |z_i - z_j| \quad (4-5)$$

The pair wise (PW) metric can be rewritten

$$PW = \frac{1}{\bar{\gamma}} \left[\frac{1}{n} \sum_{i=1}^n (\gamma_i - \bar{\gamma})^2 \right]^{\frac{1}{2}} \quad (4-6)$$

Where

$$\bar{\gamma} = \frac{1}{N} \sum_{i=1}^N \gamma_i \quad (4-7)$$

For a perfectly uniform distribution of points $\gamma_1 = \gamma_2 = \dots = \gamma_n = \bar{\gamma}$ so the $PW=0$. Small values of PW mean that the result is close to uniformly distributed.

4.4.3 Circumscription Metric (CM)

It was noted by Tahernezahadiani (2012) that in problems where solution points were clustered, the pair-wise metric may not give a reliable indication of how well distributed the points are. This is illustrated in Figure 4.4. It is clear to see that the element set in in Figure 4.4-b is more diverse and better distributed than that of Figure 4.4-a, but the values pair-wise metric for both cases are equal to zero, this is the case when the pair-wise metric fails to indicate the diversity difference between two sets.

To solve the drawback of the pair-wise distance-based diversity metric, Tahernezahadiani proposed a diversity indicator based on pair-wise with some modification.

This metric based on the principle of encompassing the data points in the population with in a circle or sphere. Once the radius, r_i , of a circle or sphere is found, the points lying on it (i.e. those defining it) are removed from the population and the process is repeated until no further circle/sphere can be defined (i.e. there is only one or no points remaining).

This metric used a monotonic logarithmic function. This function is indicated in the equation below:

$$\text{Circumscription Metric} = \frac{\log(1+100*(1+C+\sqrt{R}))}{\log(1+100)} \quad (4.8)$$

Where C= number of circles/spheres generated

$$R = \sum r_i$$

A higher value of this metric means better distribution of points in design space or objective space.

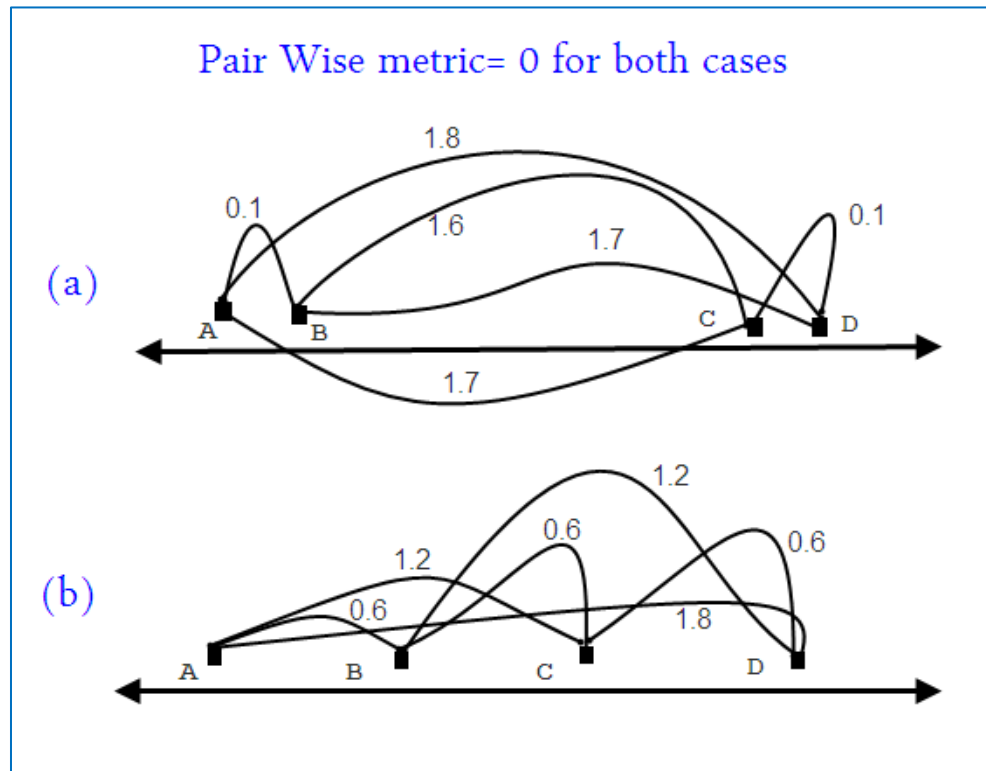


Figure 4-4: Comparison of traditional diversity indicators

4.4.4 Performance of distribution metrics

Five different cases of point distribution are used to compare the performance of Circumscription Metric (CM) and Pair-Wise metric (PW). These cases are the following:

- Case 1 uniform distribution of points Figure 4.5.
- Case 2 modifying the uniformly distributed points by moving the red point from ($x = 4.77, y = 4.77$) to ($x = 4.5, y = 4.5$) Figure 4.6.
- Case 3 grouping the points in 6 clusters Figure 4.7.
- Case 4 uniform distribution in two dimensions Figure 4.8.
- Case 5 grouping the points in 4 clusters Figure 4.9.

The value of these two metrics for the five cases studied is presented in Table 4.1. It is clear to see from this table and Figures 4.5 to 4.9, the comparison between case-1 and case-2 indicates that the two metrics give the same indication that case-1 is better than case-2 with highest CM and lowest value of PW metric, but the comparison between case-2 and case-3 shows that case-2 is the best regarding CM metric and is the worst regarding the PW metric. By comparing these results with graphical distribution of the points it is clear the CM is more realistic than PW and the latter metric fails to assess the uniformity of the distribution, especially when there is clustering in points. This situation normally happens when Genetic algorithms are used as subsequent populations are generated from individuals with only small areas of the design space.

Another comparison is done between case-4 and case-5 which indicates that case-4 is more uniform than case-5 regarding the CM metric, whilst the PW indicates that case-5 is the ideal case with a value of $PW=0$. The graphical distribution shows a different scenario in that case-4 is more uniformly distributed than case-5. In conclusion, the CM metric has a better ability than the PW metric to indicate the distribution of points, especially when there is a clustering in the points.

Table 4-1: Results of comparison between Circumscription and Pair-wise Metrics

Case study	Circumscription Metric (CM)	Pair-Wise Metric (PW)
Case 1	1.4892	0
Case 2	1.4887	0.111808
Case 3	1.4883	3.14E-16
Case 4	1.5641	0.173829
Case 5	1.5235	0

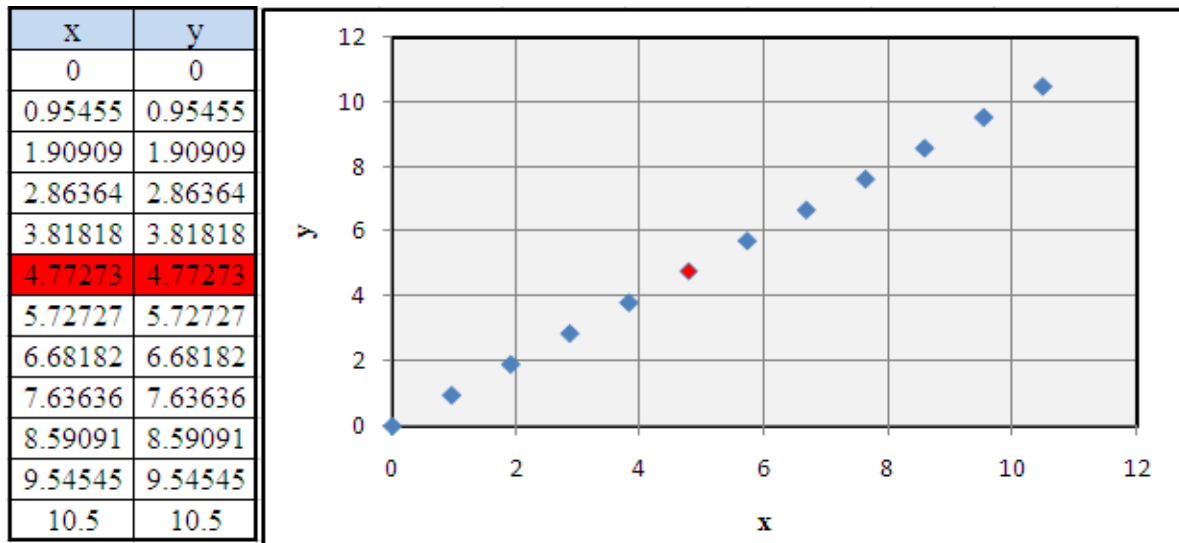


Figure 4-5: Case 1: regular distributions of points

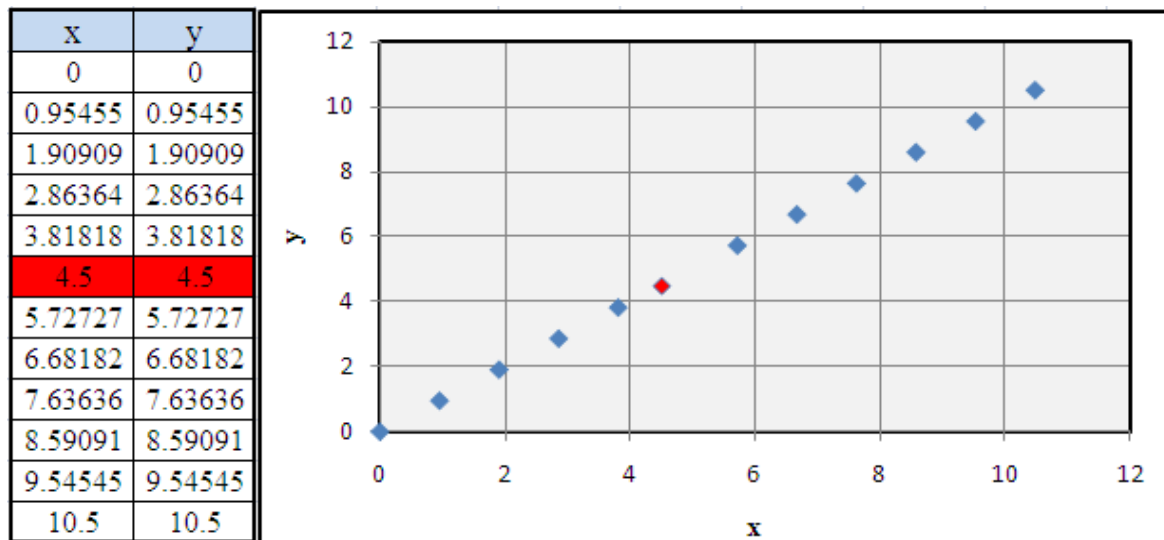


Figure 4-6: Case 2: regular distribution of points with changing the position of red points

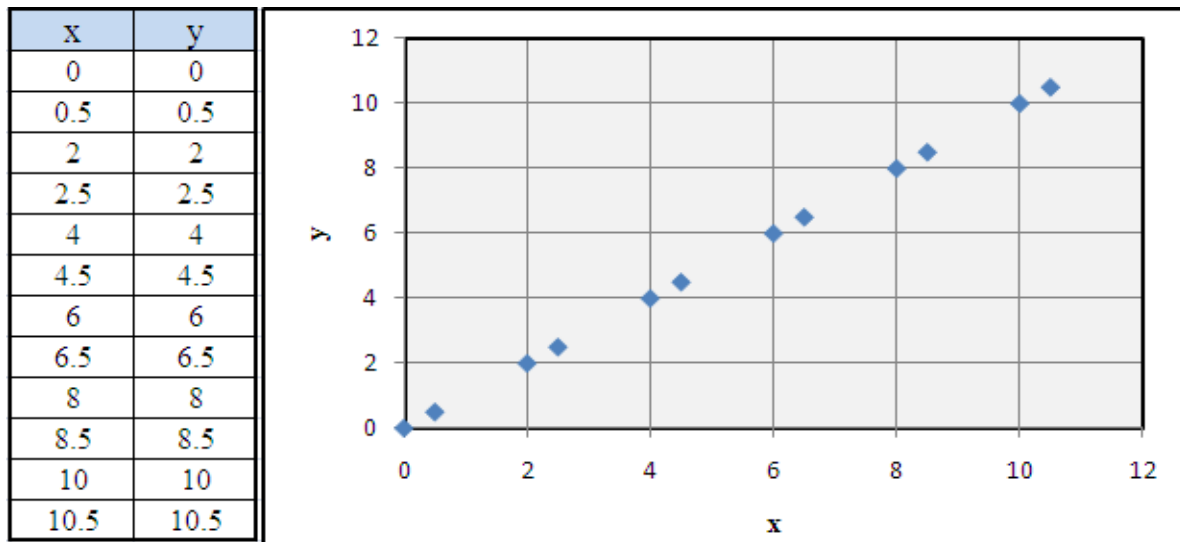


Figure 4-7: Case 3: cluster distribution of points

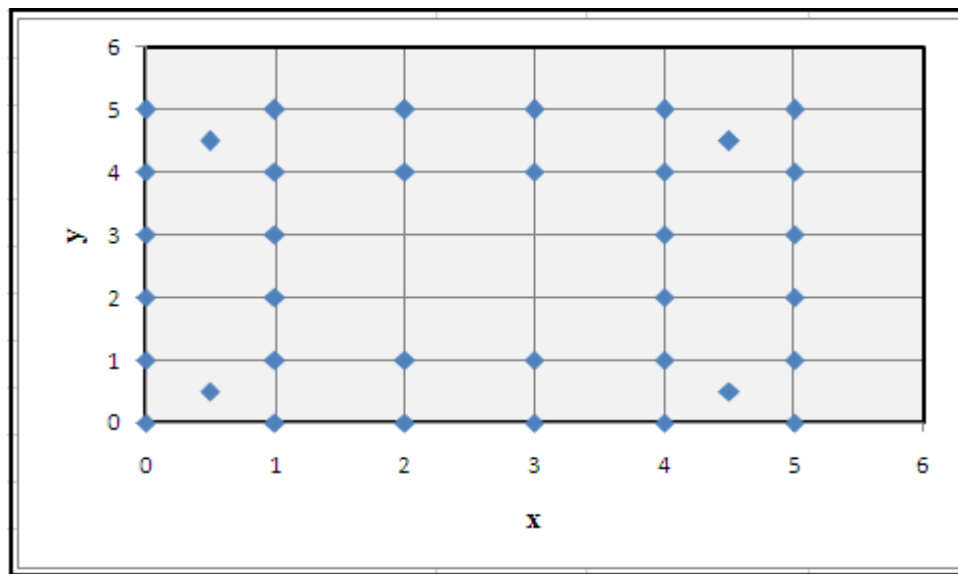


Figure 4-8: Case 4: moderate clustering of points in two dimensions.

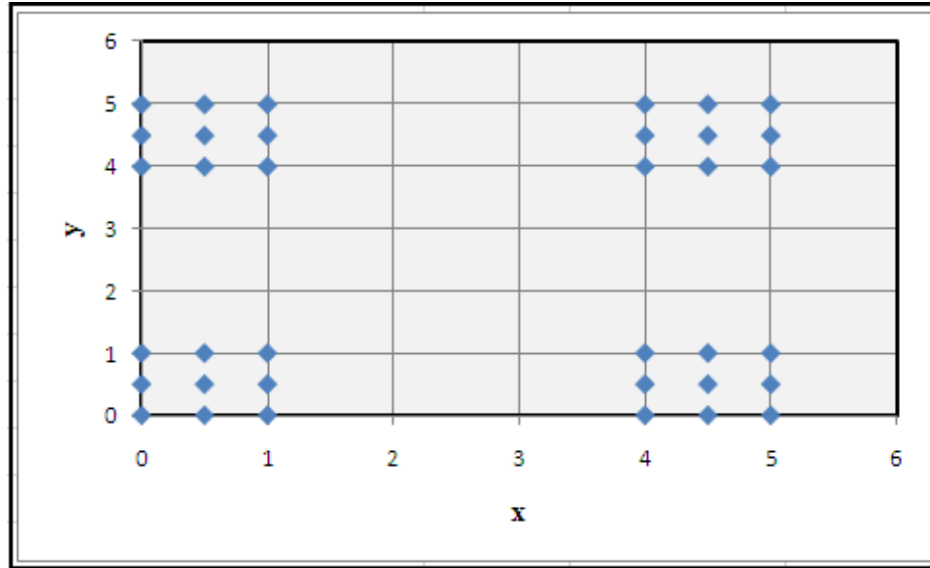


Figure 4-9: Case 5: heavy clustering of points in two dimensions.

4.5 Lifting Arm Case Study

The concept of the composition of simple pin-connected members is the basis of this example. This permits the structure to be analysed according to the simple theories of equilibrium presented by Newton as well as the recognised rules of trigonometry. Numerous structures have been planned like this since the design computations may be conducted speedily and economically.

The design of a pin-connected structure in this example - that is, a lifting arm - is going to be optimised. The objective of the optimisation will be to maximise both the range of the movement of the arm and the mass being lifted by the arm. This optimisation example was originally presented by Ghurbal (2003). To maintain consistency with that presentation, the inverse of both the force and movement criteria will be minimised. As stated above, this problem was chosen as an example of a problem with complex constraints in addition to multiple criteria (i.e. it has similar characteristics to the weld geometry problem).

4.5.1 Optimisation problem formulation

Figure 4-10, illustrates the structure where d_1 and d_2 represent the main design factors. Variation of both these values can be made based on the place of the pin-connection. The parameter restrictions for these variables for any experiment point t are indicated as follows:

$$0.1 \leq d_{1,t} \leq 0.9 \qquad 0.1 \leq d_{2,t} \leq 0.9 \qquad (4-9-a)$$

As the ram's length, d_3 can be altered by raising the arm up or dropping it down; consequently, the angle θ is transformed. Hence, it is noteworthy to state that d_3 interrelates with θ . The bounds on d_3 and θ in terms of any trial point t are as follows:

$$d_{3,min}^0 \leq d_{3,t} \leq d_{3,max}^0 \quad \theta_{min}^0 \leq \theta_t \leq \theta_{max}^0 \qquad (4-9-b)$$

Where $d_{3,min}^0$, and $d_{3,max}^0$ are respectively the real substantial minimum and maximum lengths that d_3 can adapt when the ram is not attached to the system. In addition θ_{min}^0 , and θ_{max}^0 , are correspondingly, the minimum and maximum angles that the arm can move to without the ram being connected.

The bounds selected for Equation 4-9 in this case:

$$0.2 \leq d_{3,t} \leq 0.4 \qquad 10^\circ \leq \theta_t \leq 170^\circ \qquad (4-10)$$

Thus, for each trial point t that fulfils the parameter bounds, the required length of d_3 is computed for θ_{min}^0 , and θ_{max}^0 , Therefore,

$$\bar{d}_{3min,t} = \sqrt{d_{1,t}^2 + d_{2,t}^2 - 2d_{1,t}d_{2,t}\cos\theta_{min}^\circ} \qquad (4-11)$$

$$\bar{d}_{3max,t} = \sqrt{d_{1,t}^2 + d_{2,t}^2 - 2d_{1,t}d_{2,t}\cos\theta_{max}^\circ} \qquad (4-12)$$

Where the required length of d_3 is $\bar{d}_{3min,t}$ if the minimum angle is to be acquired; in addition $\bar{d}_{3max,t}$, is the necessary length of d_3 if the maximum angle is to be achieved.

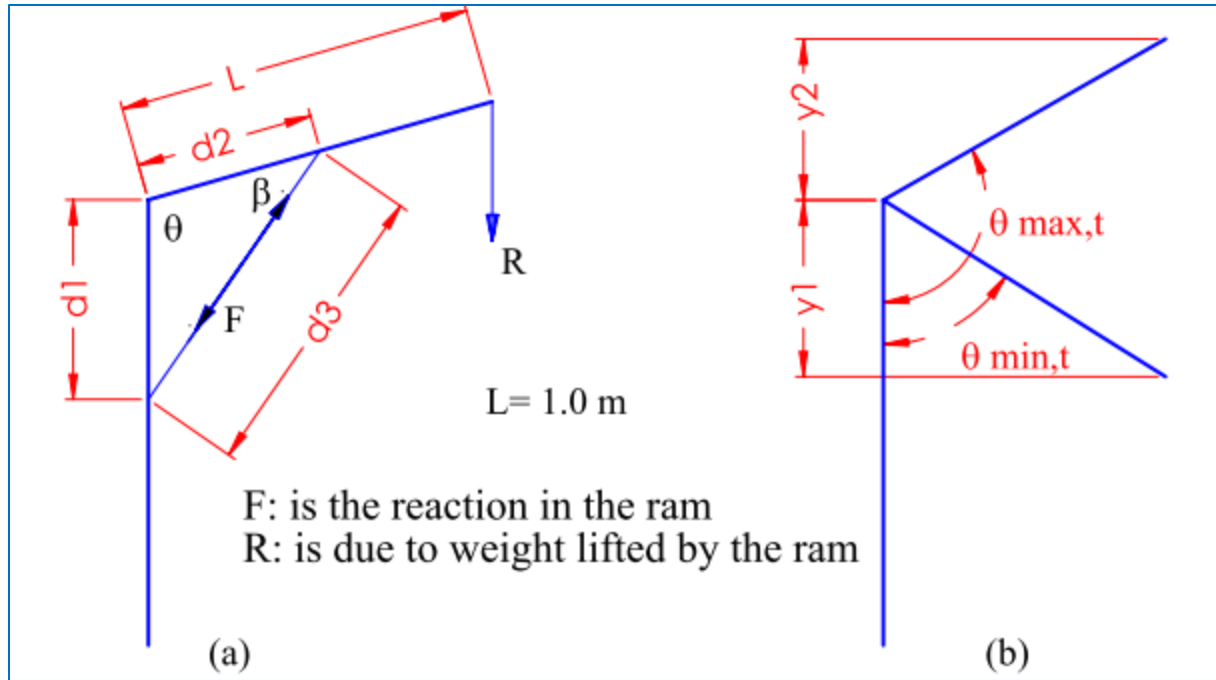


Figure 4-10: Schematic diagram of the lifting arm (Ghurbal, 2003).

There will be various conditions where the ram is not capable of being sufficiently long or short. Consequently, this called for the development of two functional constraints where a trial point t that fulfils the parameter bounds is re-checked to examine its viability. The following two equations illustrate these functional constraints:

$$g_1 \equiv \bar{d}_{3min,t} < d_{3max}^0 \quad (4-13)$$

$$g_2 \equiv \bar{d}_{3max,t} > d_{3min}^0 \quad (4-14)$$

A point is considered as feasible when it meets the constraints in Equations 4-10, 4-13 and 4-14. As the value of d_3 interrelates with θ some restrictions have to be checked at this phase to guarantee that the bounds of d_3 and θ in Equation 4-10 are not breached. For a trial point t that meets parameter and functional constraints, the minimum length of the ram is represented as follows:

$$d_{3min,t} = \begin{cases} \bar{d}_{3min,t} & \text{if } \bar{d}_{3min,t} \geq d_{3min}^0 \\ d_{3min}^0 & \text{otherwise} \end{cases} \quad \begin{matrix} \text{condition (a)} \\ \text{condition (b)} \end{matrix} \quad (4-15)$$

Where $\bar{d}_{3min,t}$ is shown as computed in Equation 4-11. The minimum angle $\theta_{3min,t}$ for the same trial point t is illustrated as follows:

$$\theta_{3min,t} = \begin{cases} \theta_{min}^0 & \text{if } \bar{d}_{3min,t} \geq d_{3min}^0 & \text{condition (a)} \\ \bar{\theta}_{min,t} & \text{otherwise} & \text{condition (b)} \end{cases} \quad (4-16)$$

Where

$$\bar{\theta}_{min} = \cos^{-1} \left[\frac{d_{1,t}^2 + d_{2,t}^2 - d_{3min}^0}{2d_{1,t} d_{2,t}} \right] \quad (4-17)$$

Notice that Equation 4-15 (condition a) interrelates with Equation 4-16 (condition a). In addition, Equation 4-15 (condition b) interrelates with Equation 4-16 (condition b). In a similar fashion, the maximum length of the ram for the trial point t can be described as:

$$d_{3max,t} = \begin{cases} d_{3max}^0 & \text{if } d_{3max}^0 < \bar{d}_{3max} & \text{condition (a)} \\ \bar{d}_{3max} & \text{otherwise} & \text{condition (b)} \end{cases} \quad (4-18)$$

Where, \bar{d}_{3max} is computed from Equation 4-12.

The maximum angle $\theta_{max,t}$ for the same trial point t is shown as follows:

$$\theta_{3max,t} = \begin{cases} \bar{\theta}_{max,t} & \text{if } d_{3max}^0 < \bar{d}_{3max,t} & \text{condition (a)} \\ \theta_{max}^0 & \text{otherwise} & \text{condition (b)} \end{cases} \quad (4-19)$$

And $\bar{\theta}_{max,t}$, can be computed as:

$$\bar{\theta}_{max,t} = \cos^{-1} \left[\frac{d_{1,t}^2 + d_{2,t}^2 - d_{3max}^0}{2d_{1,t} d_{2,t}} \right] \quad (4-20)$$

Determining the criteria values is possible once the maximum and minimum angles and ram lengths have been calculated, as discussed earlier. The criteria applied in order to evaluate this structure are the weight that can be taken at the maximum height as well as the total vertical height that can be travelled by the lifting arm.

Since the weight that can be dealt with by the structure is to be optimised, the first criterion can be described as:

$$C_{R,t} = \frac{1}{R_t} = \frac{L \sin \theta_{max,t}}{d_{2,t} F \sin \beta_t} \quad (4-21)$$

The vertical distance passed through by the lifting arm represents the other criterion for the structure.

With the aim of maximising, Y_t the second criterion can be described as follows:

$$C_{Y,t} = \frac{1}{Y_t} = \frac{1}{L (\cos \theta_{min,t} - \cos \theta_{max,t})} \quad (4-22)$$

As stated above, to maintain consistency with the work of Ghurbal (2003), the two criteria, $C_{R,t}$ and $C_{Y,t}$ will be minimised.

4.5.2 Single generation optimisation

To gain understanding of this optimisation problem, a number of single generation optimisation runs were carried out with increasing population size. This would then provide a baseline against which the results from runs using genetic algorithms could be compared. In each case the population was generated from Sobol sequence.

The first run was conducted with 16 trial points, to allow the results to be readily presented and in order to test that the optimisation routine was functioning properly.

Table 4-2 shows the trial point data for this run, and Figure 4-11 (a-1), shows the parameter domain and Figure 4-11 (a-2), the criteria domain.

The trial points are categorised as either Pareto or non-feasible points in Table 4-2. It is noteworthy that the criteria were not computed for the non-feasible points due to the fact that functional restrictions were breached. For this reason, it is not possible for the structure of the lifting arm to be formed.

An instance that demonstrates how the values of $d_{3min,t}$ and $\theta_{3min,t}$ are employed can be clarified in trial point 1. For purposes of elaboration, the calculated value of $\bar{d}_{3min,t}$ for the minimum angle of the structure, i.e. 10° , is lower than the d_3 constraint (achieved by Equation 4-10). Thus, the value of $d_{3min,t}$ is positioned to the least length of the ram; i.e. $d_{3min}^0 = 0.2$ m, and the angle $\theta_{min,t}$ is computed according to this value. Equation 3 is re-checked once more for the new value of $\theta_{min,t}$ i.e. 22.78 , and is discovered to be adequate. In the same way, the calculated length of the ram, $\bar{d}_{3max,t}$ at $\theta = 170^\circ$ causes it to be longer than the highest possible length of 0.4 m.

Table 4-2: Optimisation routine output for 16 trial points generated by Sobol

Trial point	d_1 (m)	d_2 (m)	$\bar{d}_{3min,t}$ (m)	$\bar{d}_{3max,t}$ (m)	$\theta_{min,t}$ (deg)	$\theta_{max,t}$ (deg)	c_{Yt} (N ⁻¹)	c_{Yt} (m ⁻¹)	
1	0.506	0.506	0.0883	1.0088	22.78	46.53	4.273	0.0012	Pareto
2	0.303	0.303	0.0528	0.604	38.52	82.55	1.532	0.0033	
3	0.71	0.71	0.1237	1.4136	16.20	32.74	8.39	0.0006	
4	0.405	0.608	0.2208	1.009	10.00	40.64	4.424	0.0013	
7	0.608	0.405	0.2208	1.009	10.00	40.64	4.424	0.0013	
9	0.76	0.456	0.3216	1.2115	10.00	25.43	12.24	0.0009	
11	0.557	0.252	0.3117	0.8069	10.00	40.42	4.475	0.0022	
12	0.252	0.354	0.1142	0.6041	33.50	80.66	1.489	0.0034	
13	0.659	0.76	0.1598	1.4137	13.98	31.73	8.347	0.0006	
14	0.456	0.151	0.3082	0.6046	10.00	59.24	2.113	0.0045	
15	0.862	0.557	0.3278	1.4139	10.00	21.55	18.28	0.0006	
16	0.481	0.481	0.0838	0.9582	24.00	49.14	3.855	0.0013	
5	0.811	0.202	0.6136	1.0102	10.00	170.00	N/A	N/A	Unfeasible
6	0.202	0.811	0.6136	1.0102	10.00	170.00	N/A	N/A	
8	0.354	0.862	0.517	1.2121	10.00	170.00	N/A	N/A	
10	0.151	0.659	0.5109	0.8077	10.00	170.00	N/A	N/A	

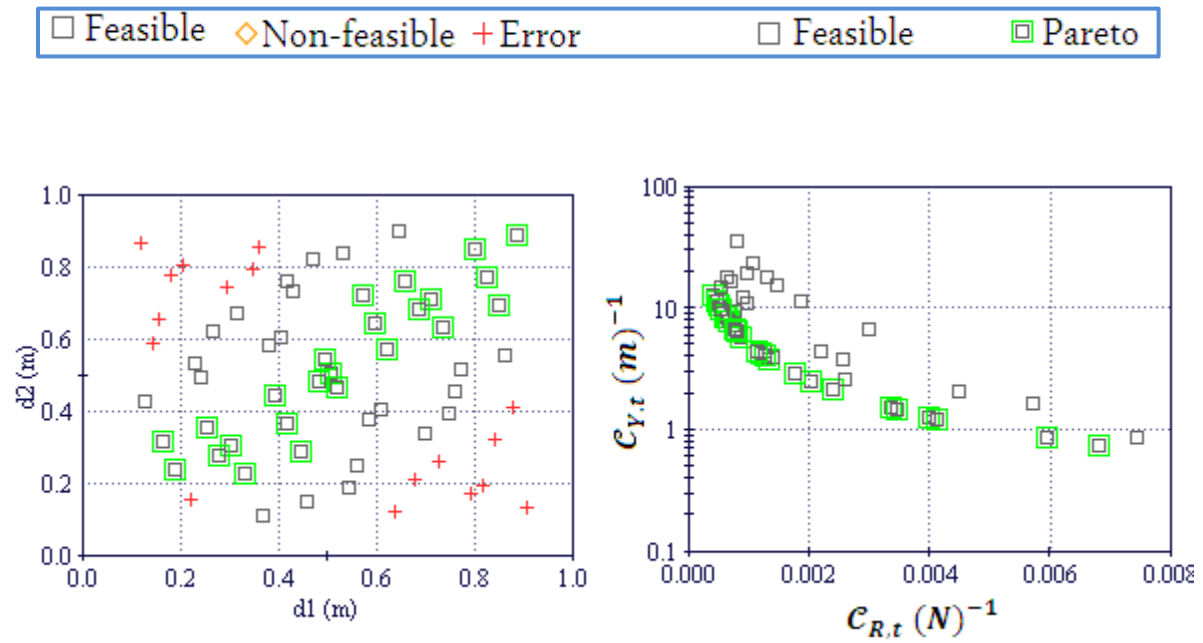
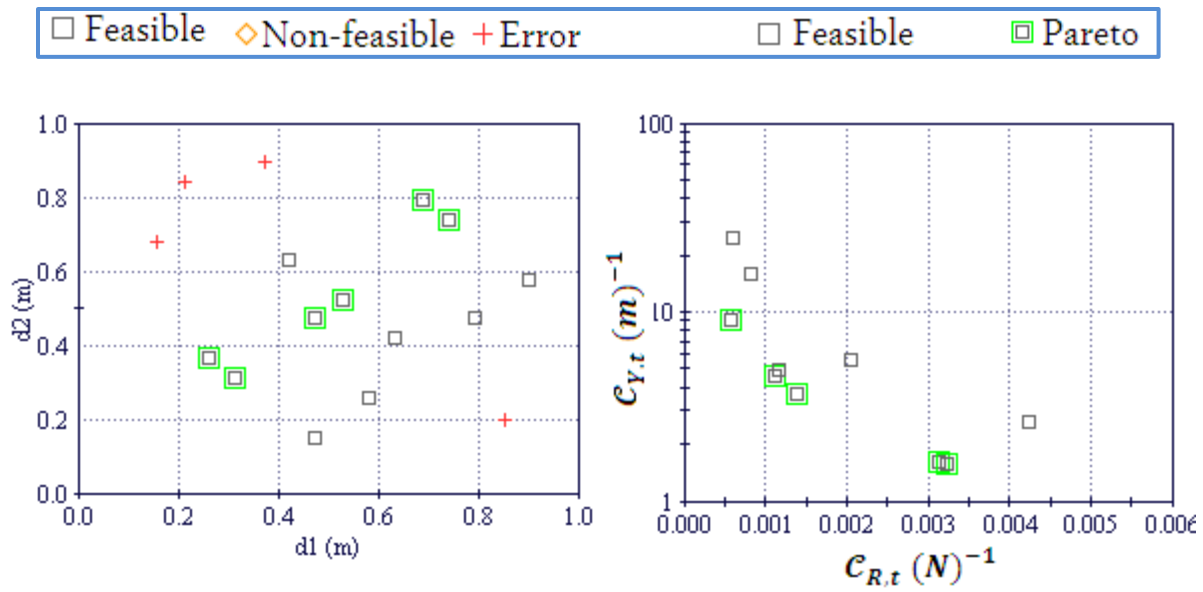
Consequently, the ram length is adjusted to 0.4 m and $\theta_{max,t}$ is found to be equal to 46.66°. Accordingly, the structure of the lifting arm can be present with this trial point. When a feasible arrangement of the design parameters is found the criteria are possible values are also calculated.

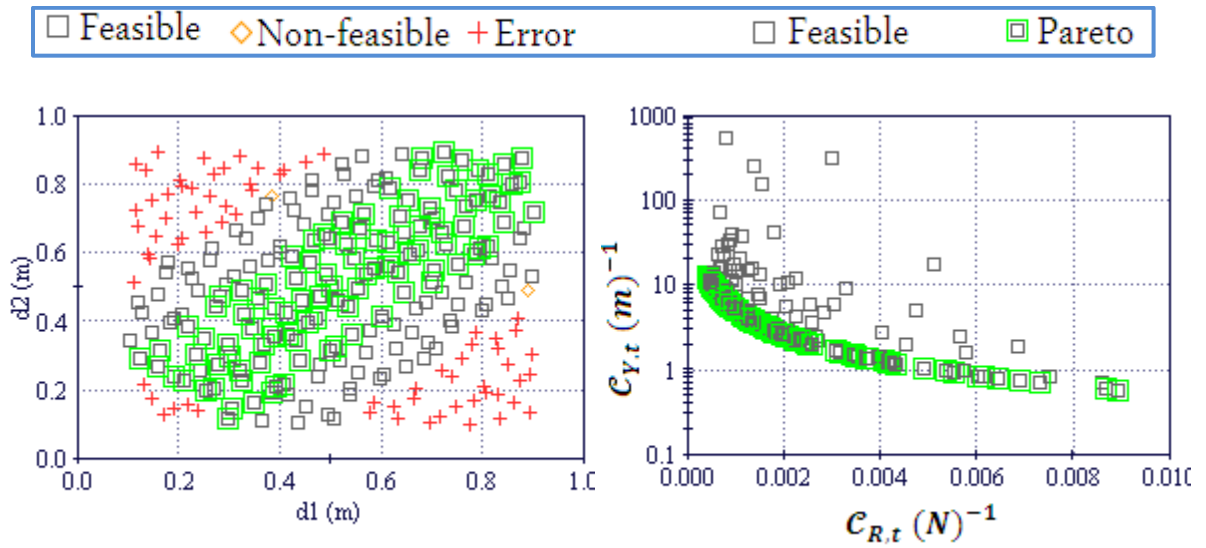
The purpose of employing this process is to establish the Pareto design parameters for the problem.

The Pareto set convergence is investigated as the optimisation problem is re-checked by increasing the numbers of the trial points created by the Sobol sequence. In this section, Figure 4.11 clearly illustrates that the number of the trial points were raised in the form of 2^4 , 2^6 , 2^8 and 2^{10} . As a result, when applying the Sobol sequence each design parameter was split into K equal parts and a point was positioned within each of these parts. It would be simpler to say that K^R trial points were created where K indicates the number of parts in every design parameter and R signifies the number of design parameters; that is, in this case $R=2$. From Figure 4.11, it is clear that the uniformity in the domain should be point independent and the convergence of solution is possible without using high trail points.

In the parameter fields illustrated in Figure 4.11 it is obvious that the optimisation routine recognises three different patterns of solutions; that is, Pareto, feasible or non-feasible. The feasible set is positioned between the Pareto and the non-feasible sets. It should be noted that the points identified as ‘error’ are members of the non-feasible set. They have been identified as ‘error’ solutions because the criteria calculation could not be completed (e.g. when the required ram length is outside the permitted range).

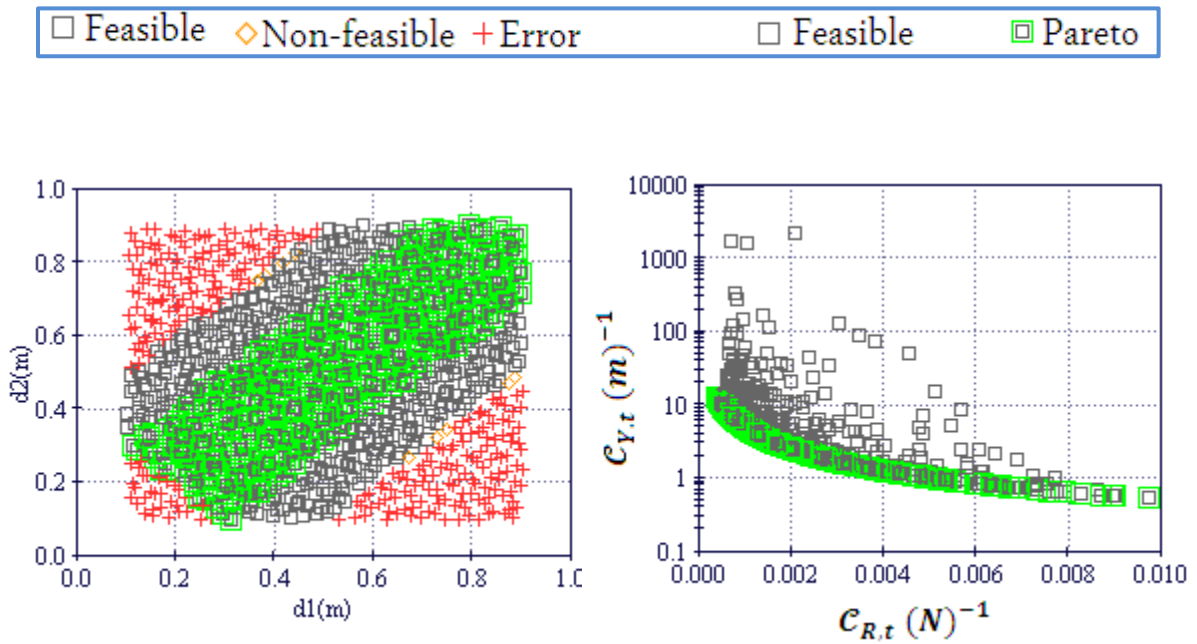
In Figure 4.11, the three sets of solutions (Pareto, feasible and non-feasible solution) are clearly formed as a result of increasing the number of trail points. When a high number of trail points (i.e. 1,024) are used, the Sobol sequence produces clustered points. However, a high number of trail points is not required in this problem.





c-1) Variable space for 256 trial points

c-2) Objective space for 256 trial points



d-1) Variable space for 1024 trial points

d-2) Objective space for 1024 trial points

Figure 4-11: Variable and objective space for trial points generated by the Sobol sequence.

4.5.3 Optimisation with Genetic Algorithms.

As mentioned earlier in this Chapter Six population-based optimisation algorithms are implemented for the comparison and effectiveness of each. The parameters used during this simulation are the following:

- Number of input variables: 2
- Number of objective functions: 2
- Size of initial population: 64.
- Initial population seeding method: Sobol sequence.
- Number of generations: 5, 10, 15, 20, 25, 30.

The input and output parameters are linked with an Excel spread sheet as indicated in Figure 4.12. Figure 4.13 indicates the designed problem in modeFRONTIER.

input var		output var		
d1	0.506	Rt	Rt	832.117
d2	0.506	Ht	Ht	0.234342046
L	1	θ_{max}	th_m_max	0.812695053
F	1300	θ_{min}	th_m_min	0.397876143
		θ^{max}	thetamax	0.812695053
		θ^{min}	thetamin	0.397876143
		$X3^{min}$	X3min	0.088201612
		$X3^{max}$	X3max	1.008149034
		X3maxt	X3maxt	0.4
		sinbeta	sinbeta	0.918570612
			fRt	0.001201754
			fHt	4.267266667
		$\theta_{max}(deg)$		46.56399656
		$\theta_{min}(deg)$		22.79662378

Figure 4-12: Definition of input and output design parameters in lifting arm problem.

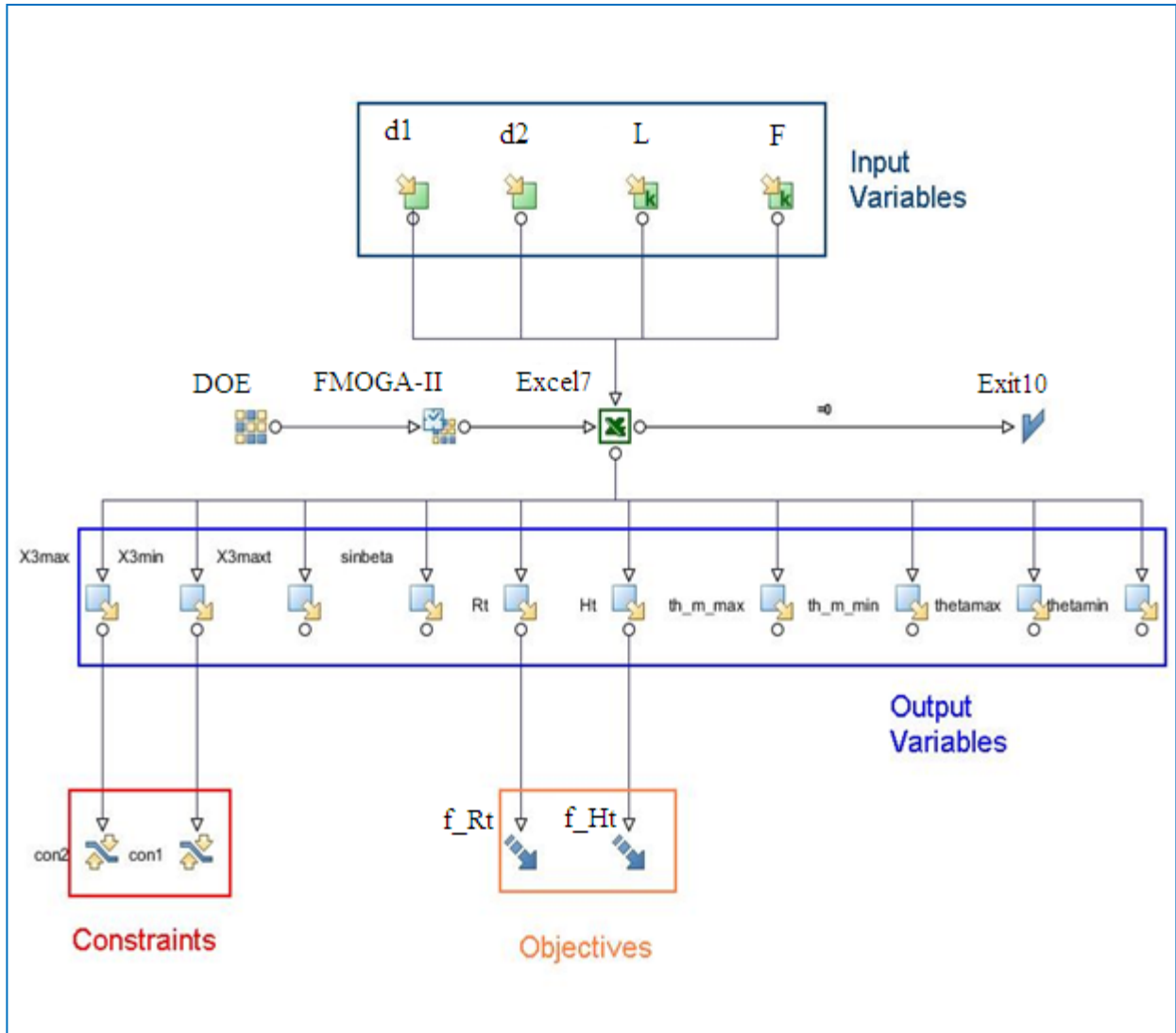


Figure 4-13: Lifting arm project within modeFRONTIER.

Included in Table 4-3 are the obtained values of Pair-Wise metric which indicates the spread of the solution in design space. The graphical representation of these results is also presented in Appendix A. It is clear that the FMOGA-II has the smallest value of this metric in all generations and gives a uniform distribution over all the Pareto set in design space (see Fig. Appendix A).

With an increase in the number of generations the performance of some algorithms, such as MOSA and MOPSO, regarding Pair-Wise metric does improve, whereas there is no significant improvement in the performance of ARMOGA and MOGA algorithms.

The values of Hit-Rate metric are presented in Table 4-3 and Figure 4-14 for 5, 10, 15, 25 and 30 generations with 64 Sobol sampling points. From these results it can be seen that the FMOGA-II outperformed all other algorithms in 5, 10 and 15 generations, whereas in 20, 25 and 30 generations the ARMOGA is the best one.

In all algorithms in this study an increase in the number of generations leads to a reduction in the value of this metric except in the case of ARMOGA. This increase indicates an improvement in the behaviour of this algorithm.

Figures 4-15, 4-16, and 4-17 present the graphic results for all algorithms in criteria space. Using this graphical representation of the Pareto optimal solution curve found by the six methods it is possible to compare their performance. In 5 generations (Figure 4-15), it is evident that the FMOGA-II and NSGA-II algorithms performed equally well. They both displayed a better distribution of the Pareto set than the other methods. These other methods performed well at the mid-section, but found very few Pareto points at the extremes.

For 10 generations (Figure 4-16) it can be seen that the FMOGA-II and NSGA-II methods showed a uniform distribution of Pareto optimal solution; however, other methods gave a poor distribution at one end of the curve, such as MOSA, or at both ends like MOPSO, MOGA-II and ARMOGA.

In the last case, with 15 generations (as shown in Figure 4-17), it should be noted that FMOGA-II outperformed the other methods. NSGA-II and MOSA performed reasonably well while there was a poor distribution of the Pareto optimal solution at the extremes for the remaining methods.

Regarding the Circumscription Metric, it can be seen from the results in the table that the FMOGA-II outperformed the other algorithms, but a significant thing that can be noticed in this case is that for a high number of generations (i.e. 25 and 30) there is a only a slight difference in the value of this metric regarding the FMOGA-II, MOGA-II and NSGA-II algorithms. However, the performance of algorithms will improve by increasing the number of generations.

Table 4-3: Performance measures of MOGA-II, ARMOGA, NSGA-II, MOSA, FMOGA-II and MOPSO algorithms for lifting arm problem showing the values of Hit-Rate (HR%) Pair-Wise (PW) and Circumscription (CM) Metrics

	MOGA-II	ARMOGA	NSGA-II	MOSA	FMOGA-II	MOPSO
5- Generations						
HR% metric	36.9	62.2	40	42.5	76.4	50.3
PW-metric	0.609865	0.6972	0.47689	0.46314	0.31481	0.48767
CM-metric	1.6136	1.6122	1.6125	1.5977	1.6389	1.6265
10- Generations						
HR% metric	39.7	69.4	37.8	46.1	78.3	48.8
PW-metric	0.482801	0.71993	0.46759	0.40901	0.2843	0.39145
CM-metric	1.6503	1.6378	1.6515	1.6386	1.6508	1.6506
15- Generations						
HR% metric	38.9	74.2	34.9	43.1	77.2	44.2
PW-metric	0.405288	0.66359	0.36282	0.31941	0.21325	0.33227
CM-metric	1.6609	1.6491	1.662	1.6499	1.6622	1.6618
20- Generations						
HR% metric	37.1	74.8	28.4	42.7	65.2	40.8
PW-metric	0.352981	0.71009	0.41545	0.21764	0.16917	0.25128
CM-metric	1.6506	1.6491	1.6505	1.6383	1.6508	1.6496
25- Generations						
HR% metric	34	74.8	28.4	40.8	60.8	37.7
PW-metric	0.332608	0.69856	0.31624	0.21464	0.14559	0.21412
CM-metric	1.6513	1.6491	1.6499	1.6497	1.6514	1.6386
30- Generations						
HR% metric	30.4	73.5	27.3	40.4	60.5	35.5
PW-metric	0.333216	0.65416	0.2809	0.15853	0.14559	0.19048
CM-metric	1.6566	1.6542	1.6505	1.6498	1.6572	1.6505

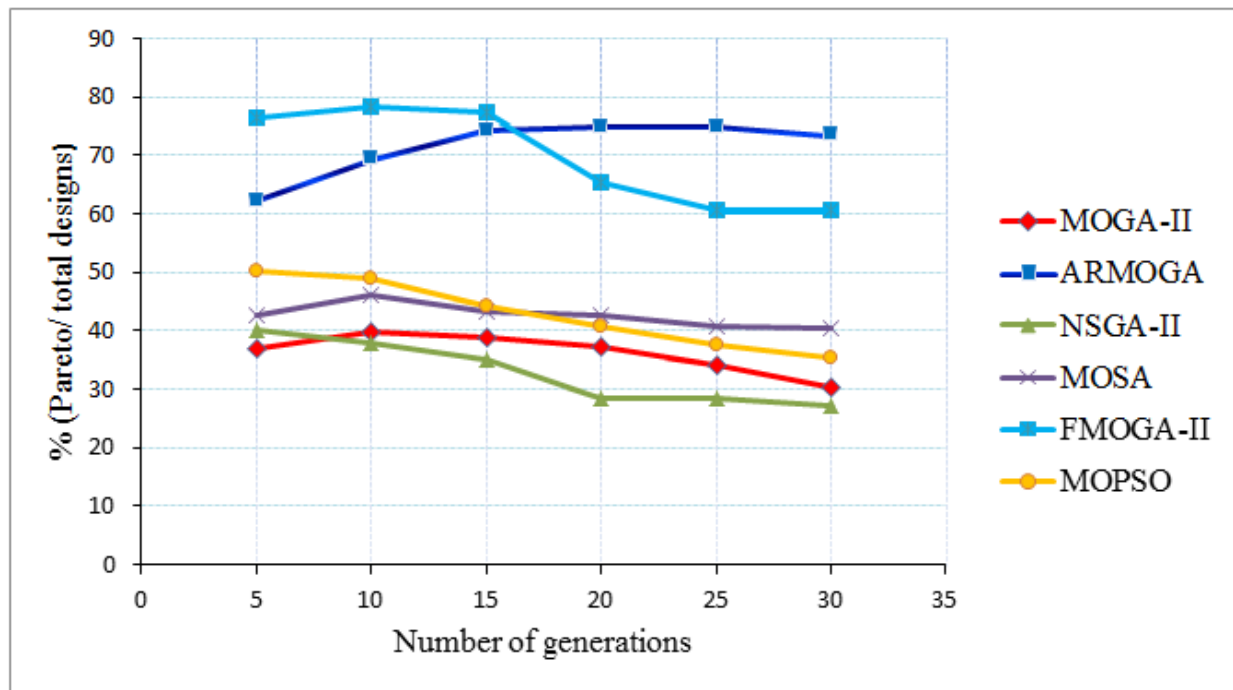


Figure 4-14: Comparison of the Hit-Rate value for 64 sampling points with different generations.

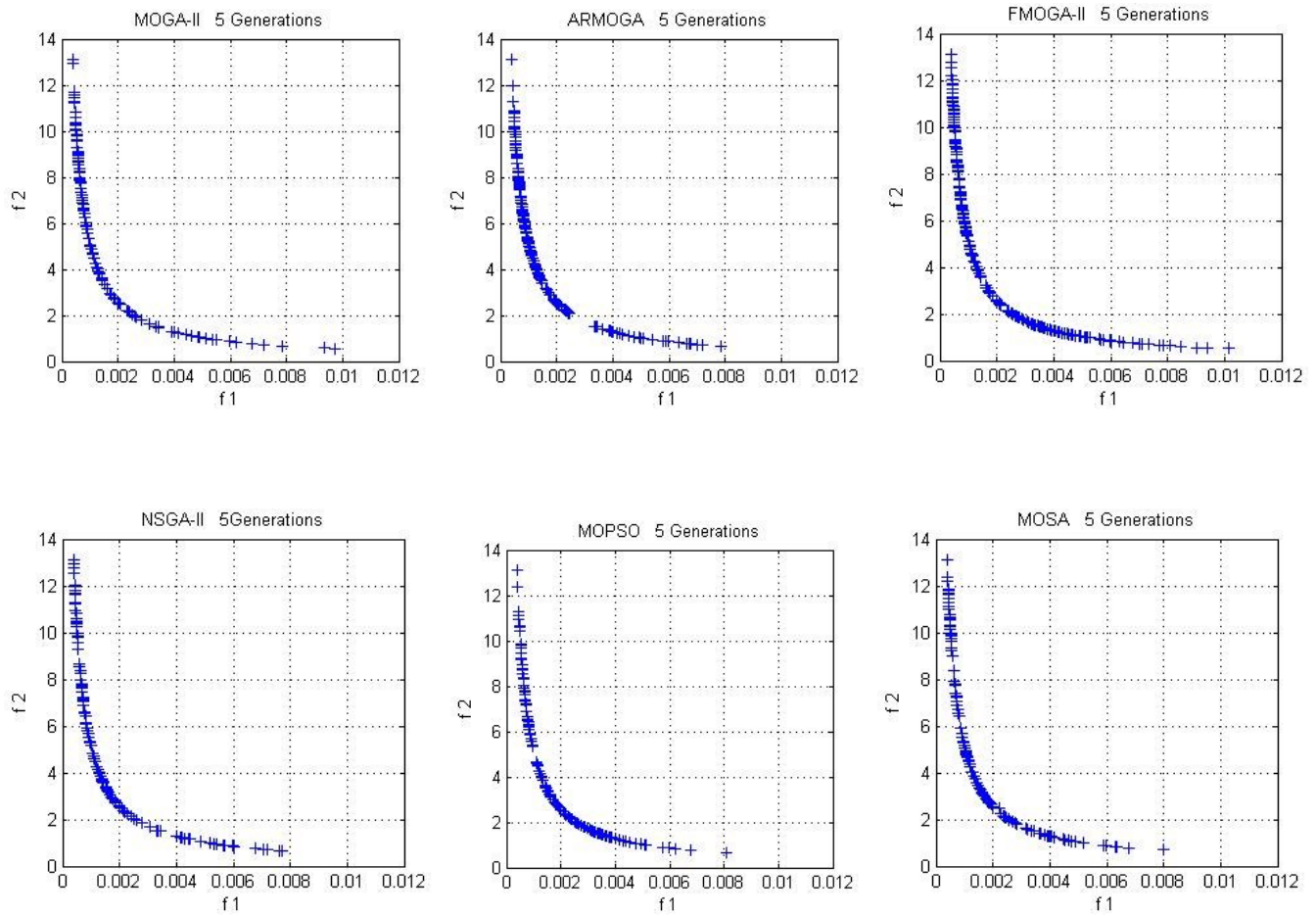


Figure 4-15: Comparison of the 5 generations results to the lifting arm problem by MOGA-II, ARMOGA, FMOGA-II, NSGA-II, MOPSO and MOSA.

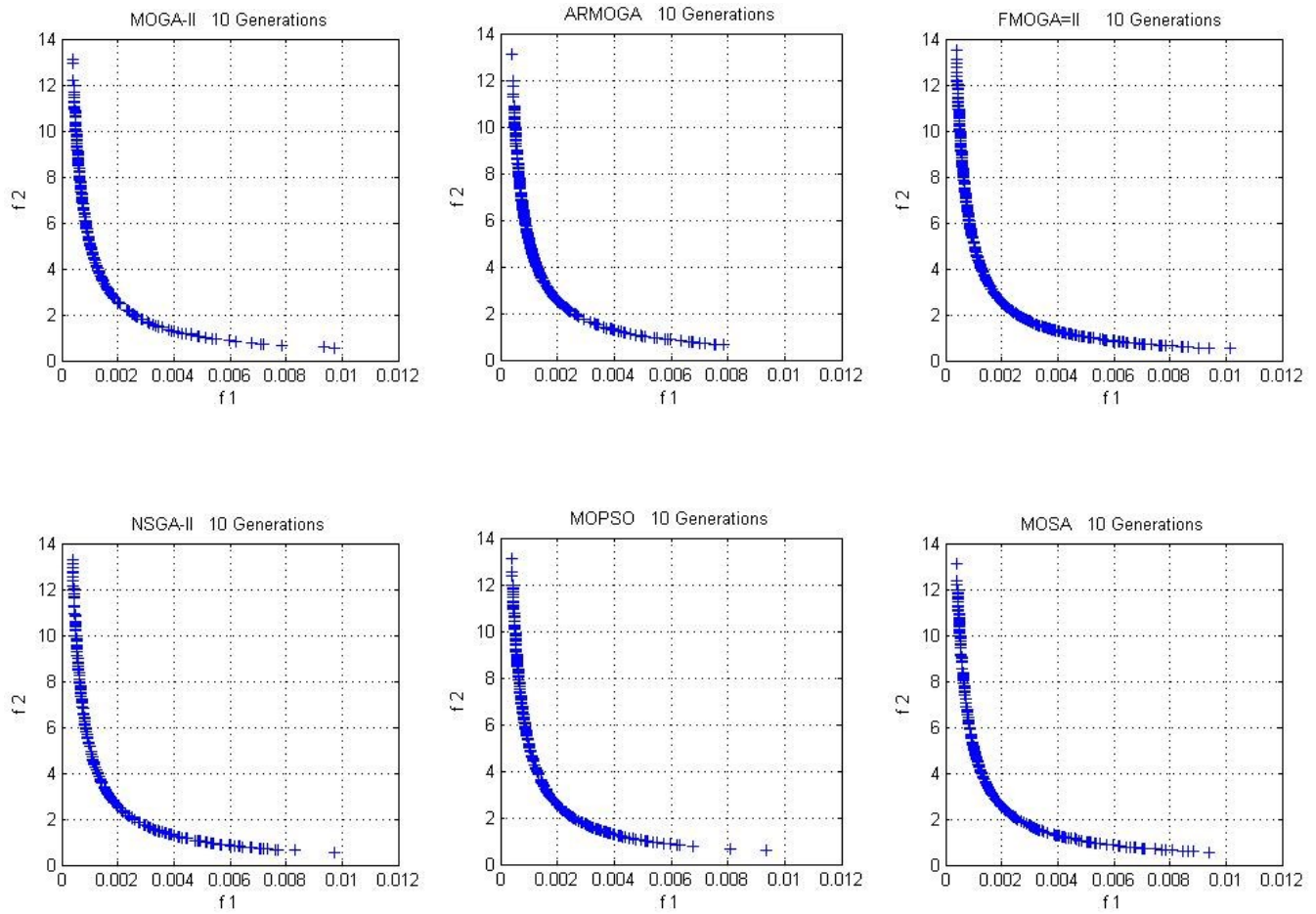


Figure 4-16: Comparison of the 10 generations results to the lifting arm problem by MOGA-II, ARMOGA, FMOGA-II, NSGA-II, MOPSO and MOSA.

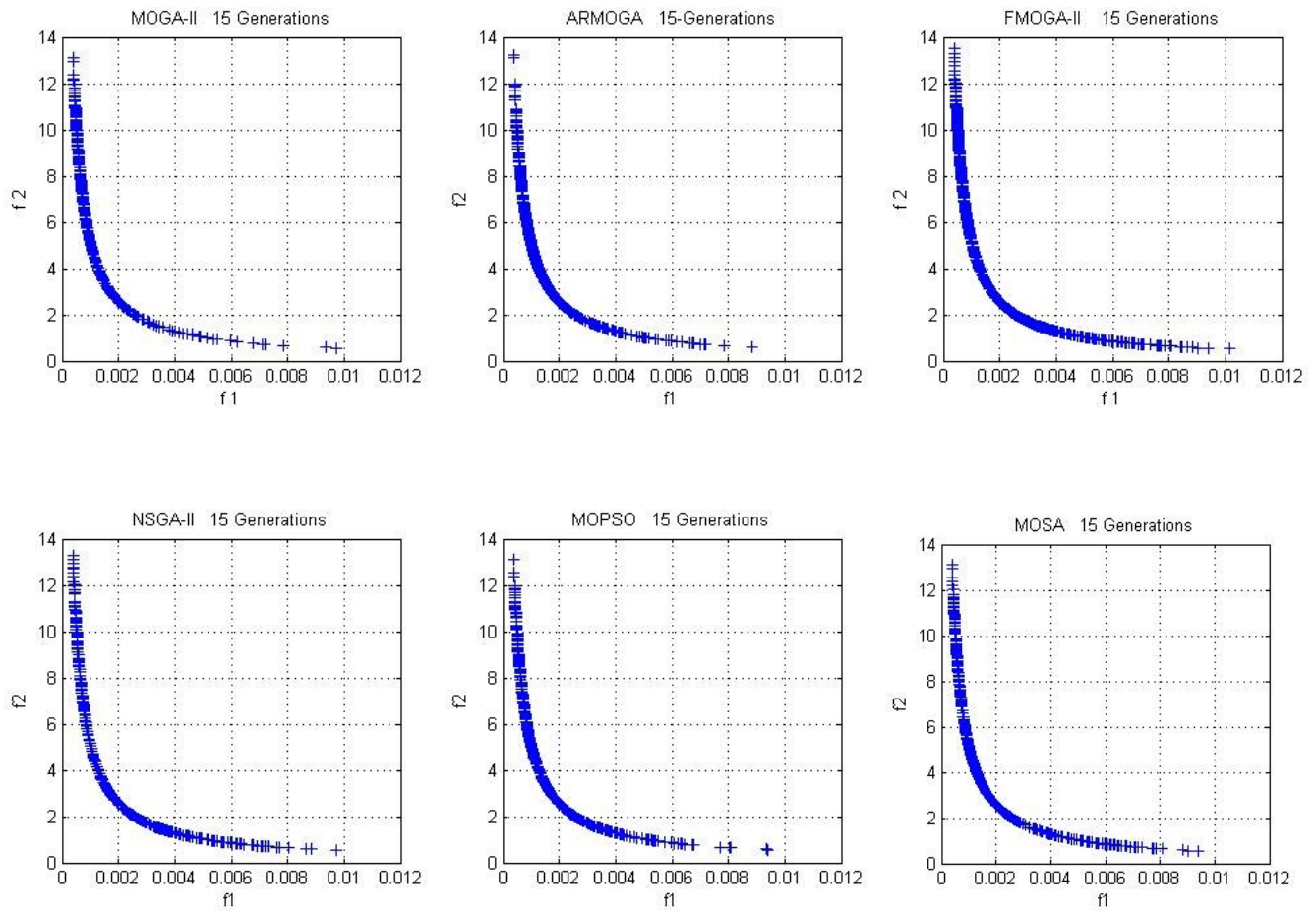


Figure 4-17: Comparison of the 15 generations results to the lifting arm problem by MOGA-II, ARMOGA, FMOGA-II, NSGA-II, MOPSO and MOSA.

4.6 Convex and Concave Case Studies

In the literature, the measurement of the performance of multi-objective optimisation algorithms is often done by applying them to two types of benchmark problems, with convex and concave solution sets. This is done with the objective of reduces the impact of the form of the solution space on the evaluation of the algorithms. There is a wide range of different problems used with varying parameters. In this study, two different benchmark problems are used. SCH and FON are widely used problems in the field of multi-objective optimisation.

The following parameters are used during this study:

- Initial population size, 100.
- Crossover probability, 0.65.
- Mutation probability, 0.1.
- Number of generations, 10.

4.6.1 SCH problem

This is a low dimensional problem suggested by Schaffer (1987).

$$SCH = Minimize(f_1, f_2) \quad (4-23)$$

$$f_1(x) = x^2$$

$$f_2(x) = (x - 2)^2$$

$$x \in [-10^3, 10^3]$$

4.6.2 FON problem

This is a problem used by Fonseca and Fleming (1998). It is characterised by having a non-convex Pareto front and nonlinear objective functions with values concentrated around $f_1, f_2 = (1, 1)$.

$$FON = Minimize(f_1, f_2) \quad (4-24)$$

$$f_1(x_1, x_2, x_3) = 1 - \exp \left[\sum_{i=1}^3 \left(x_i - \frac{1}{\sqrt{3}} \right)^2 \right]$$

$$f_1(x_1, x_2, x_3) = 1 - \exp \left[\sum_{i=1}^3 \left(x_i + \frac{1}{\sqrt{3}} \right)^2 \right]$$

4.6.3 Results and discussion

As mentioned before, in order to compare the performance of the multi-objective optimisation algorithm, the Hit-Rate metric (Section 4.4) and Circumscription Metric, which measure the extent of diversity of an approximation set, were used in objective space.

To illustrate the performance of the six algorithms, Table 4-4 shows the obtained values of the two comparison metrics. For the SCH problem it can be seen that those obtained by FMOGA-II and MOGA-II are the best regarding the circumscription metric, while the solution sets obtained by NSGA-II are the second best with the same value of Circumscription Metric.

From these results, it is concluded the FMOGA-II and MOGA-II are suitable for solving the convex problem. Also the FMOGA-II outperforms all other algorithms regarding spacing and Hit-Rate for the concave problem.

From the values of the two metrics regarding the FON problem (Table 4.4), it is clear that FMOGA-II outperforms all algorithms with the highest percentage of Hit-Rate metric and highest value Circumscription Metric also.

Figures 4-18 and 4-19 present the graphical results for all algorithms. By using this graphical representation of the Pareto optimal curve found by the performance of the six methods can be compared.

In the SCH problem Figure 4-18, it is evident that the FMOGA-II and MOGA-II algorithms performed equally well. They displayed the best distribution of the Pareto front, NSGA-II the second best; on the other side, ARMOGA gave a poor distribution at one end of the curve.

For the FON problem shown in Figure 4-19, it is clear FMOGA-II shows a uniform distribution of the Pareto optimal curve. However, other methods gave a poor distribution at one end of the curve, such as ARMOGA, MOGA-II, NSGA-II and MOPSO, or at both ends, such as MOSA.

For the SCH problem, since the number of points generated in the Pareto front is small in comparison to FON problem, see Figures 4-18 and 4-19, this will lead to a slight difference in the value of both metrics for the algorithms tested

The first conclusion that can be drawn from this study is that the FMOGA-II algorithm is the best regarding the performance metrics and graphic distribution of the Pareto front in both types of problem. Another point is that the behaviour of some algorithms is dependent on the nature of the problem being investigated.

Table 4-4: Performance measures of MOGA-II, ARMOGA, NSGA-II, FMOGA-II, MOSA and MOPSO for SCH and FON problems showing the values of Hit-Rate (HR %) and Circumscription (CM) Metric

Algorithm	Hit Rate Metric (HR [%])		Circumscription Metric (CM)	
	SCH problem	FON problem	SCH problem	FON problem
MOGA-II	2.23	9.29	1.4638	1.582
ARMOGA	1.94	6.17	1.4382	1.5818
NSGA-II	1.82	16.4	1.4635	1.5631
FMOGA-II	2.99	38.8	1.4638	1.6242
MOSA	0.533	1.3	1.3374	1.4271
MOPSO	1.02	4.23	1.4331	1.5477

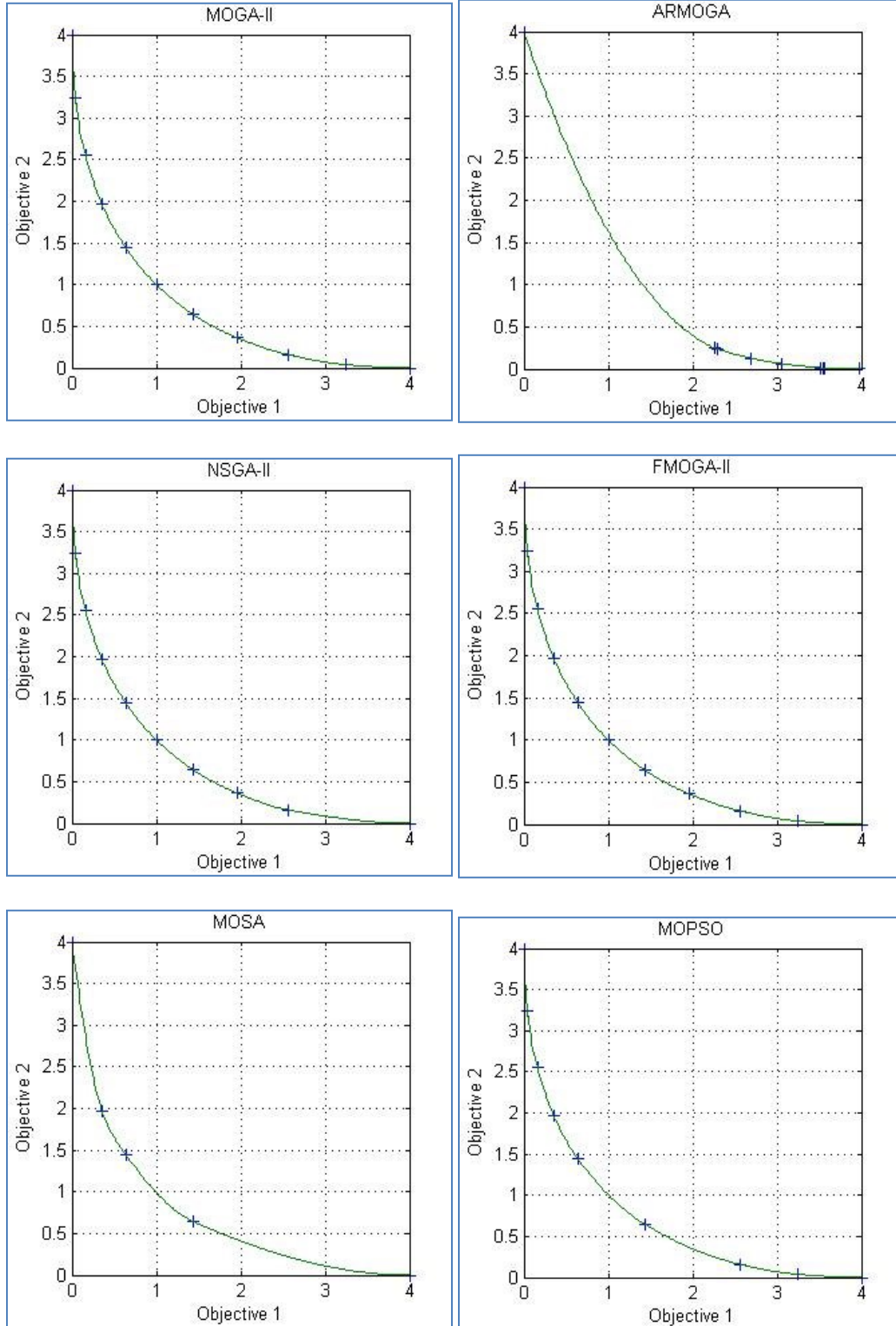


Figure 4-18: The evaluated front from MOGA-II, ARMOGA, NSGA-II, FMOGA, MOSA and MOPSO for SCH problem.

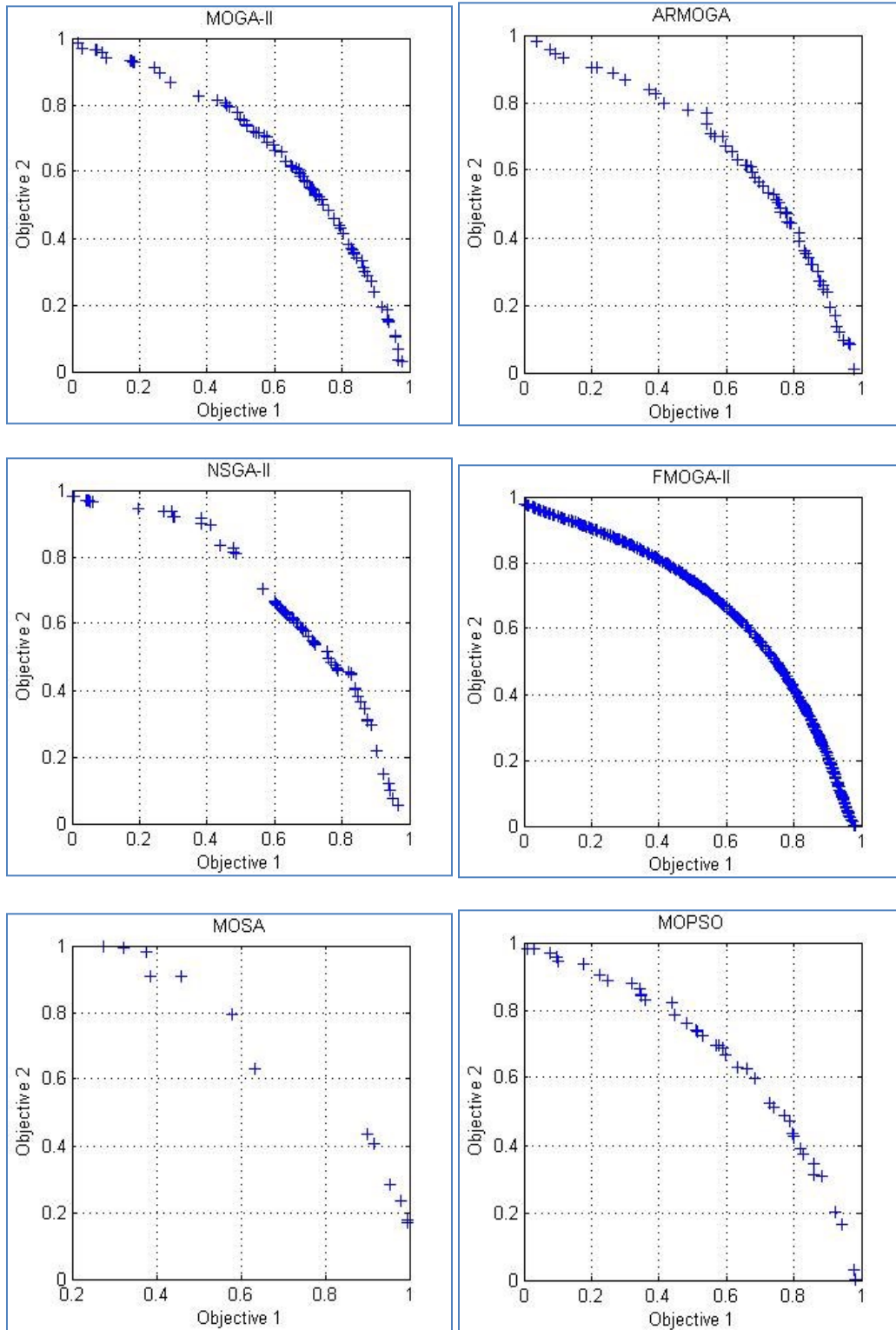


Figure 4-19: The evaluated front from MOGA-II, ARMOGA, NSGA-II, FMOGA, MOSA and MOPSO for SCH problem.

4.7 Conclusion

In this chapter, the behaviours of the six multi-objective population-based algorithms have been investigated. The testing has been carried out on examples from two different classes of problem: convex and non-convex Pareto fronts and also a practical problem with complex constraints.

The reliability of two performance metrics was investigated. It was shown that when there is a clustering of points the Pair-Wise metric (PW) fails to correctly indicate the uniformity of distribution whilst the Circumscription Metric (CM) overcomes this difficulty.

The obtained results show that in three diverse problems which have been investigated FMOGA-II outperformed the other algorithms according to the three metrics used and also qualitatively in the graphical distributions in design and objective space.

In the lifting arm problem, an increase in the number of generations improved the performance of the algorithms.

Chapter 5

Robust Optimisation

Chapter Five: Robust Optimisation

5.1 Introduction

Two different case studies have been investigated during this chapter: the first case study is the design of a fluid storage tank. This problem has a single objective function and is therefore simple to analyse. Also, the most robust design can readily be identified using closed form analysis so the example is suitable to benchmark the robust design features in modeFrontier. The second case study is the robust optimisation of the lifting arm. This example has been chosen as it features multiple design variable, functional and objective constraints. It is also an example that the research team are very familiar with and have a good understanding of the expected robust solution.

In the first problem (tank design), we are employing the philosophy of robust design in a single criteria problem by finding the value of input (design) parameters (radius, r and height h), which give smallest variance of volume (objective function) with the presence of noise or uncontrollable parameters. These parameters can be described by the manufacturing tolerances, aging and environmental effect. In other words, the resulting design is as insensitive to noise as possible and thus a robust, quality design.

In the second problem a more practical multi-criteria optimization example has been used in order to demonstrate the principles of robust design optimisation in a multi-criteria problem. In order to make the problem as simple as possible and reduce the computation time an Excel spread sheet is used to link between input and output design variables instead of finite element analysis. Three different approaches according to the initial population are performed to clarify the effect of the initial population on robust design optimisation

5.2 Case Study 1: Tank Design

To illustrate the robust optimisation of a single objective problem, the engineering problem is adapted from Ullman (1994).

The goal of this problem is to investigate the concept of robustness. Since the target is presented as a numerical value, it is easy to evaluate the product's design in relation to the target.

A product is considered as being robust if the quality measures remain high regardless of the variation in the parameters (noise) due to the uncontrollable parameters, such as manufacturing, aging and environment

By considering the design of the tank problem to hold a liquid with cylindrical shape which has an internal height (h) and internal radius (r) the volume of the tank can be expressed as:

$$V = \pi r^2 h \quad (5.1)$$

Additionally, the target is to design the tank to hold the volume as close to 4 m^3 as possible of liquid with r and h as parameters $V=4 \text{ m}^3$ as target response

$$V = 4 \text{ m}^3$$

$$r^2 h = 1.27 \text{ m}^3$$

Figure 5.1 indicates there are an infinite number of solutions to the problem. The tank at point 2 is thin and long and short and fat at point 1.

Referring back to Figure 3.4, if r and h can be considered as control parameters and they have manufacturing variation, actually they have distribution about the nominal value. The problem aim now is to reduce the dependence of the distribution of target V on design variables r and h whilst ensuring that $V=4 \text{ m}^3$ to within a given tolerance.

The source of noise factor which are difficult to control or can be controlled only at high cost can be considered as follows:

1. **Manufacturing process and tolerances:** varying from grinding, welding and machining, with tolerances from micrometres to millimetres. These tolerances will impact the actual values of r and h .
2. **Aging effect:** if the liquid stored in the tank is corrosive and with time will increase r and h by etching the inside of the tank.
3. **Environmental effect:** if the tank is manufactured and will be installed at different times with a wide range of temperatures, so r and h will vary.

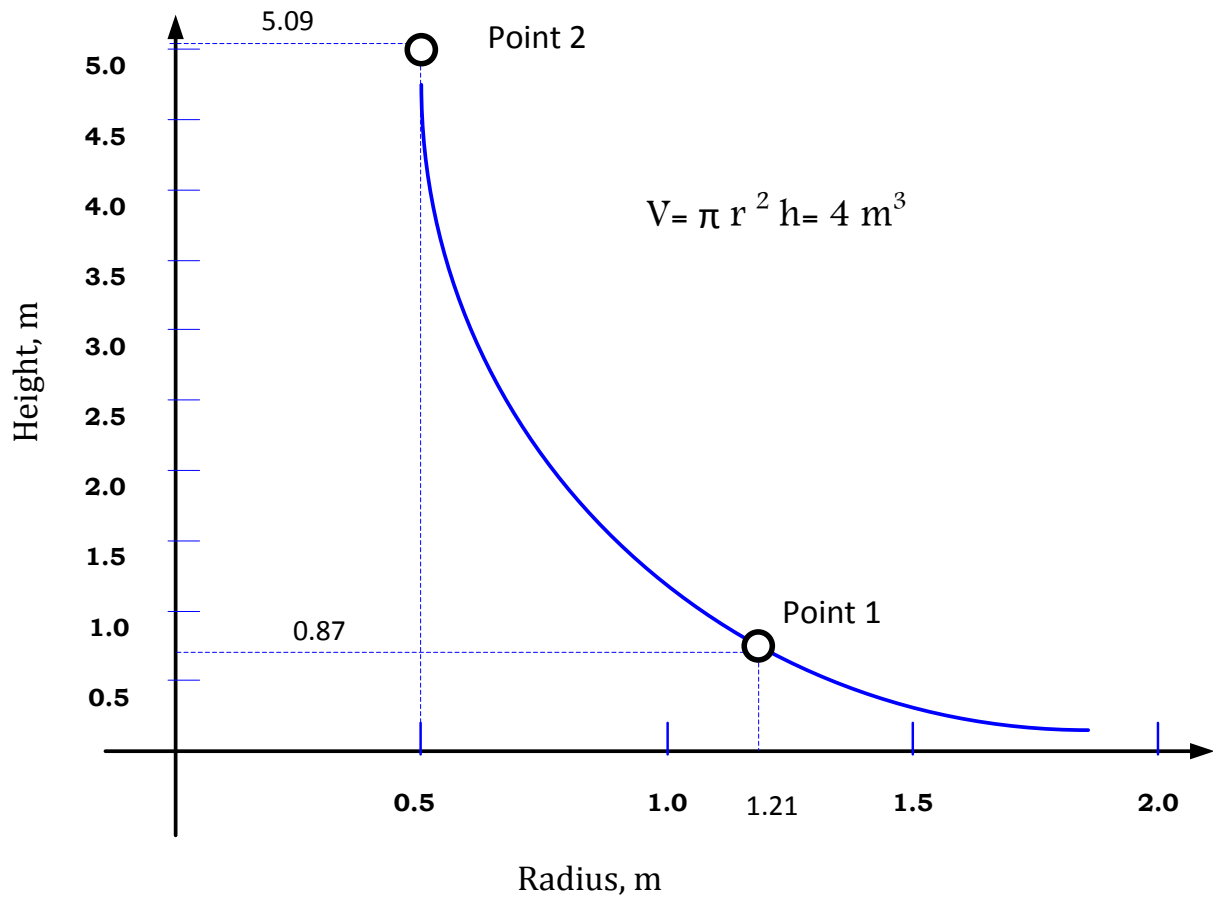


Figure 5-1: Solution for the tank problem

5.2.1 Sensitivity analysis

This is a technique for evaluating the statistical relationship of design parameters (e.g. dimension) and their tolerances in a design problem. In the tank problem, the design parameters (Radius and Height) are not linearly related to the dependent variable (Volume), as can be seen in Figure 5.1.

The functional relationship between the dependent and independent parameters can be written as:

$$F = f(x_1, x_2, x_3 \dots x_n) \quad (5.2)$$

Where F is the dependent variable (volume) and x_i are the independent variables.

The equation relating the mean value of the dependent variable \bar{F} and the independent variables \bar{x}_i can be expressed as:

$$\bar{F} = f(\bar{x}_1, \bar{x}_2, \bar{x}_3, \dots, \bar{x}_n) \quad (5.3)$$

Whilst, the standard deviation:

$$S = \left[\left(\frac{\partial F}{\partial x_1} \right) S_1^2 + \dots + \left(\frac{\partial F}{\partial x_n} \right) S_n^2 \right]^{\frac{1}{2}} \quad (5.4)$$

The mean value of the dependent variable is given by

$$\bar{V} = \pi \bar{r}^2 \bar{h} \quad (5.5)$$

Considering the specific values of r and h which satisfy the requirement ($V = 4 \text{ m}^3$): Point 1 in Figure 5.1 with $\bar{r} = 1.21 \text{ m}$ and $\bar{h} = 0.87 \text{ m}$, from equation (5.5) gives $V = 4 \text{ m}^3$. With tolerances on the design parameters of $t_r = 0.03 \text{ m}$ ($s_r = 0.01$) and $t_h = 0.15 \text{ m}$ ($s_h = 0.05$), the standard deviation of the dependent variable is given by:

$$S_v = \left[\left(\frac{\partial V}{\partial h} \right) S_h^2 + \left(\frac{\partial V}{\partial r} \right) S_r^2 \right]^{\frac{1}{2}} \quad (5.6)$$

$$\frac{\partial V}{\partial r} = 2\pi r h \quad \text{and} \quad \frac{\partial V}{\partial h} = \pi r^2 \quad (5.7)$$

So it is easy to find that $S_v = 0.239 \text{ m}^3$. By the same technique, if point 2 is taken at $r = 0.5 \text{ m}$ and $h = 5.09 \text{ m}$ the standard deviation of the volume will be equal to $S_v = 0.166 \text{ m}^3$ which is 31% smaller than at point 1.

Note that the reduction in standard deviation can be achieved by changing the nominal value and not by changing the standard deviation of the design parameters.

The design at point 2 is high-quality and more robust because the volume is always closer to 4 m^3 and that is the philosophy of robust design by finding the value of design parameters (r and h) which give smallest standard deviation on the output parameters (volume).

5.2.2 Optimisation problem parameters

As mentioned before, the problem aim is to find a robust design which gives the target volume as close to 4 m^3 as possible. The parameters of this problem are indicated in Table 5.1.

Table 5-1: General parameters of tank problem

Design Variables	Design Variables Limits	Design Variables Standard Deviation	Initial Population	Number of Generations
Radius (r)	$0.5 \leq r \leq 1.5$	0.01	100 Sobol	10 with FMOGA-II Algorithm
Height (h)	$0.5 \leq h \leq 5$	0.05		

The objective of the problem is to minimise the standard deviation of volume. The definition of this problem in modeFRONTIER software is presented in Figure 5.2.

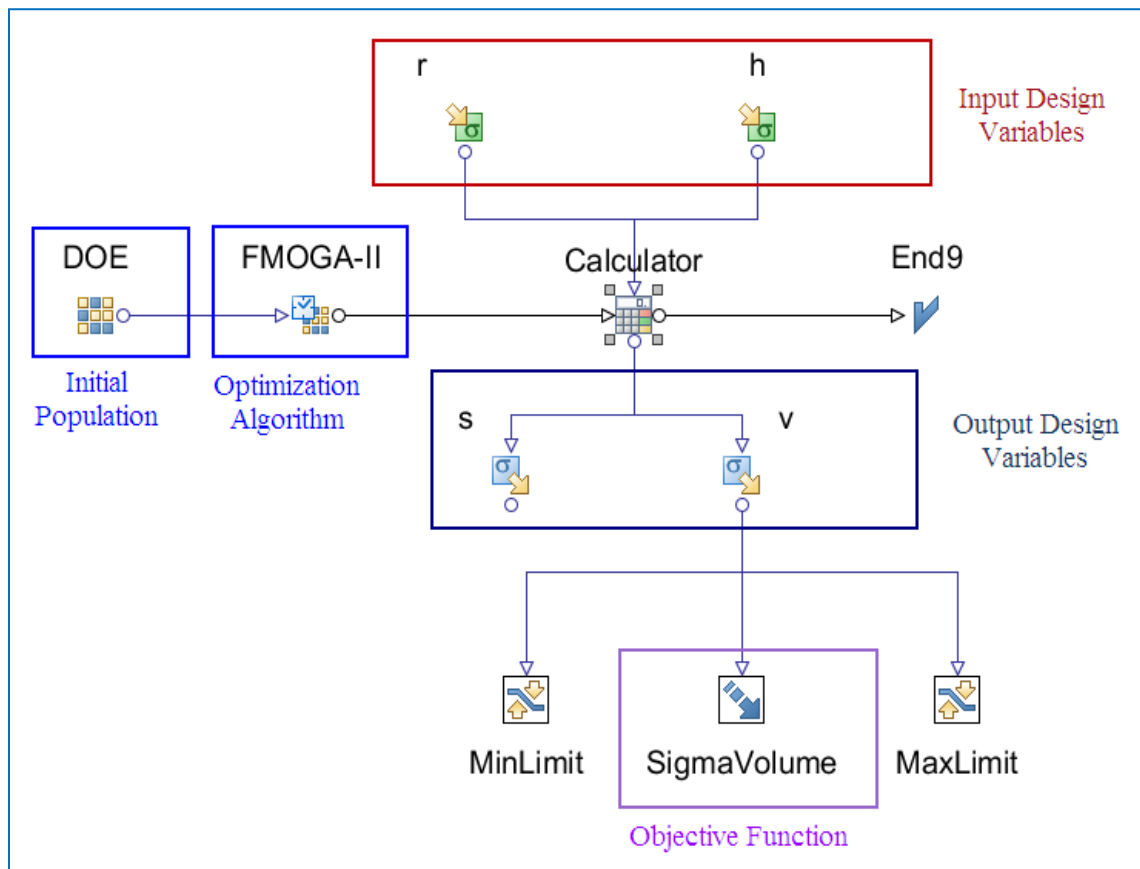


Figure 5-2: Robust optimisation of the tank problem in modeFRONTIER software

The curved line in Figure 5.1 gives an infinite number of solutions which satisfy the target $V=4 \text{ m}^3$, but it is difficult to capture these points in modeFRONTIER software. The best technique to use in this case is to consider this curve as a reference and add lower and upper bound ($\pm 20\%$, ± 10 , ± 5 , ± 2.5) surrounding this line and then searching for target solutions inside this region.

Four different cases according to the tolerance values on mean value of volume have been studied with the same initial population size and generation method and the same genetic algorithm. These cases are shown in Table 5.2.

Table 5-2: Results of tank problem

Case No.	Tolerances	Initial. Population	Generations	Min Std	No of Feasible Robust Design	No of Total Design
1	$\pm 20\%$	100 Sobol	10 FMOGA-II	0.032399	813	1000
2	$\pm 10\%$	100 Sobol	10 FMOGA-II	0.033353	853	1000
3	$\pm 5\%$	100 Sobol	10 FMOGA-II	0.033874	762	1000
4	$\pm 2.5\%$	100 Sobol	10 FMOGA-II	0.034147	798	1000

The distribution of design variables (r and h) in design space are presented in Fig. 5.3.

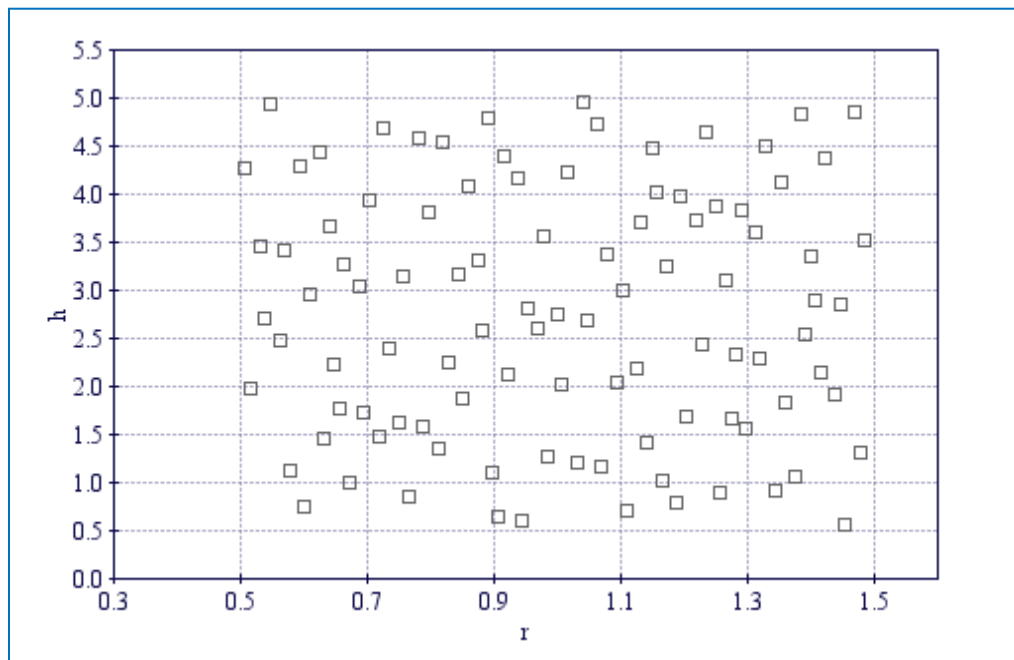


Figure 5-3: Spatial distribution of design variables in design space

In mono-objective optimisation the history chart indicates how the optimisation algorithm evolves. Additionally, it indicates the minimum value of objective function after a certain number of iterations.

The objective function (sigma volume) is plotted as a function of design ID in a robust design table for all four cases and is presented from Figure 5.4 to Figure 5.7.

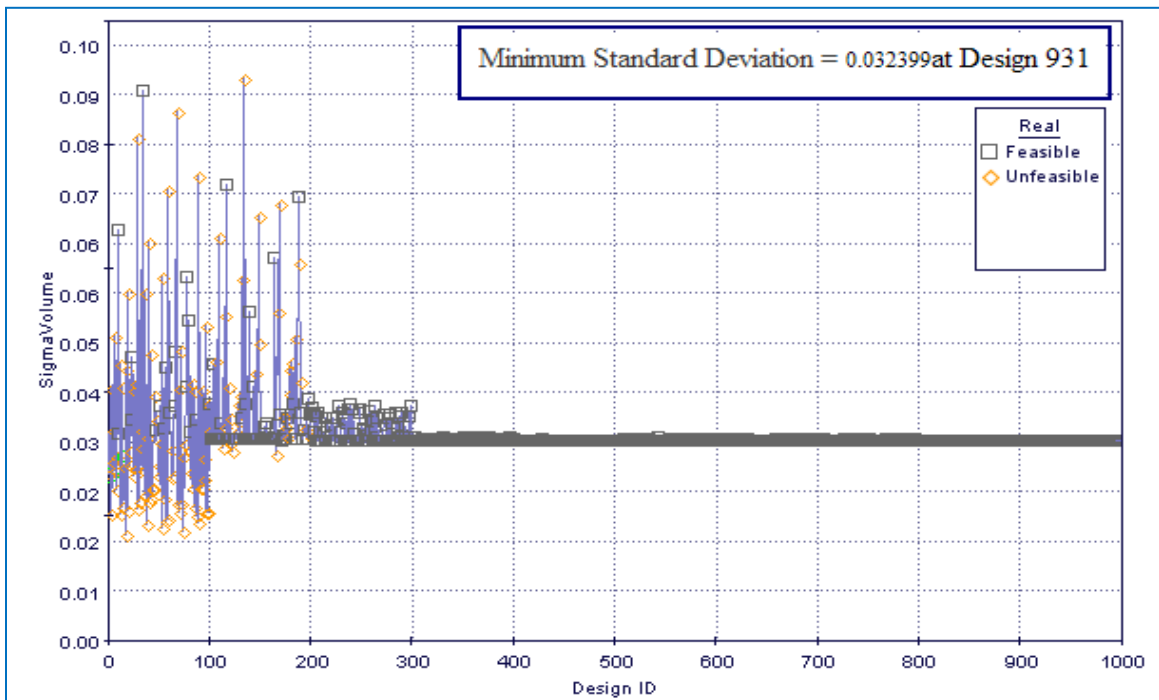


Figure 5-4: History chart of objective function with 20% tolerance

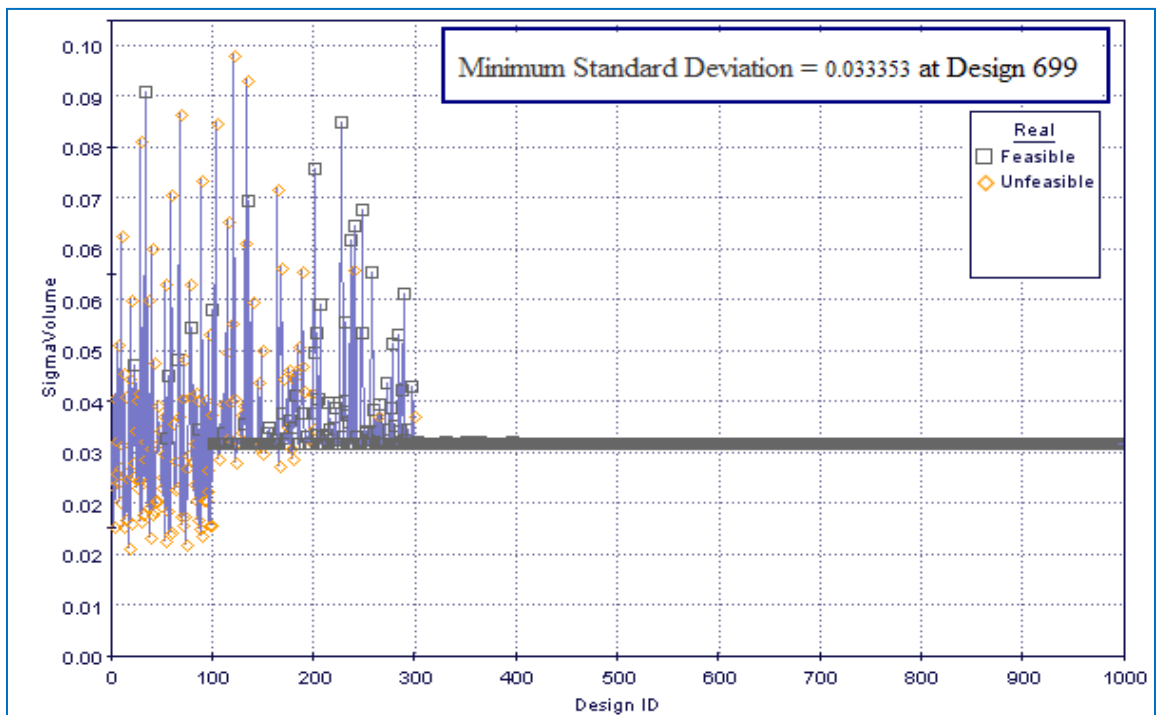


Figure 5-5: History chart of objective function with 10% tolerance

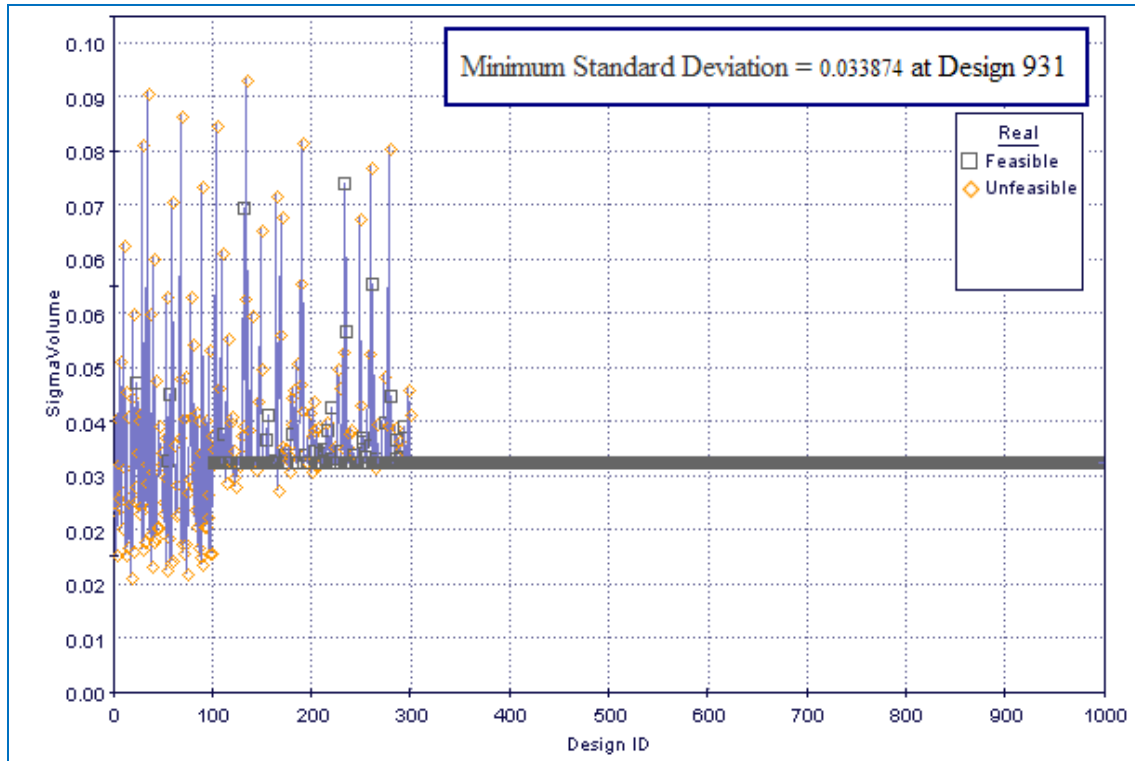


Figure 5-6: History chart of objective function with 5% tolerance

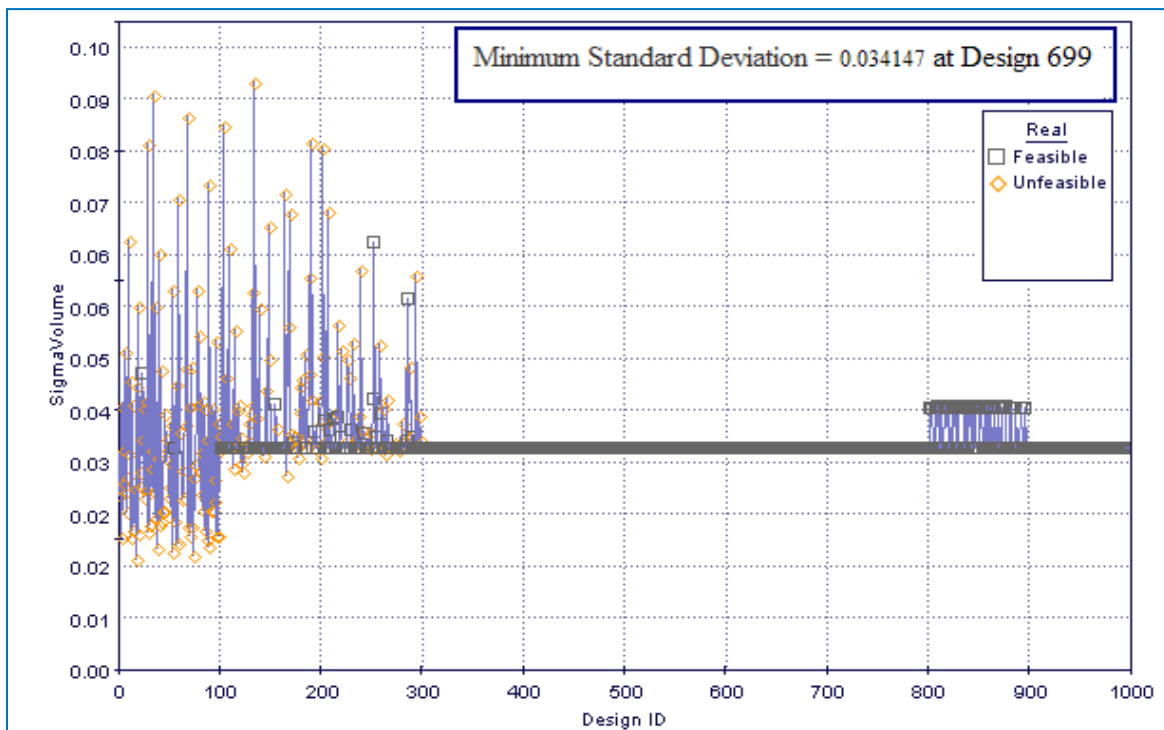


Figure 5-7: History chart of objective function with 2.5% tolerance

The results of the robust optimisation analysis can also be shown in design space (see Figures 5.8 to 5.11). In these figures the design points have been divided into five groups according to the value of standard deviation on volume.

The blue diamond group is a more robust design with the smallest value of standard deviation on the system response (volume).

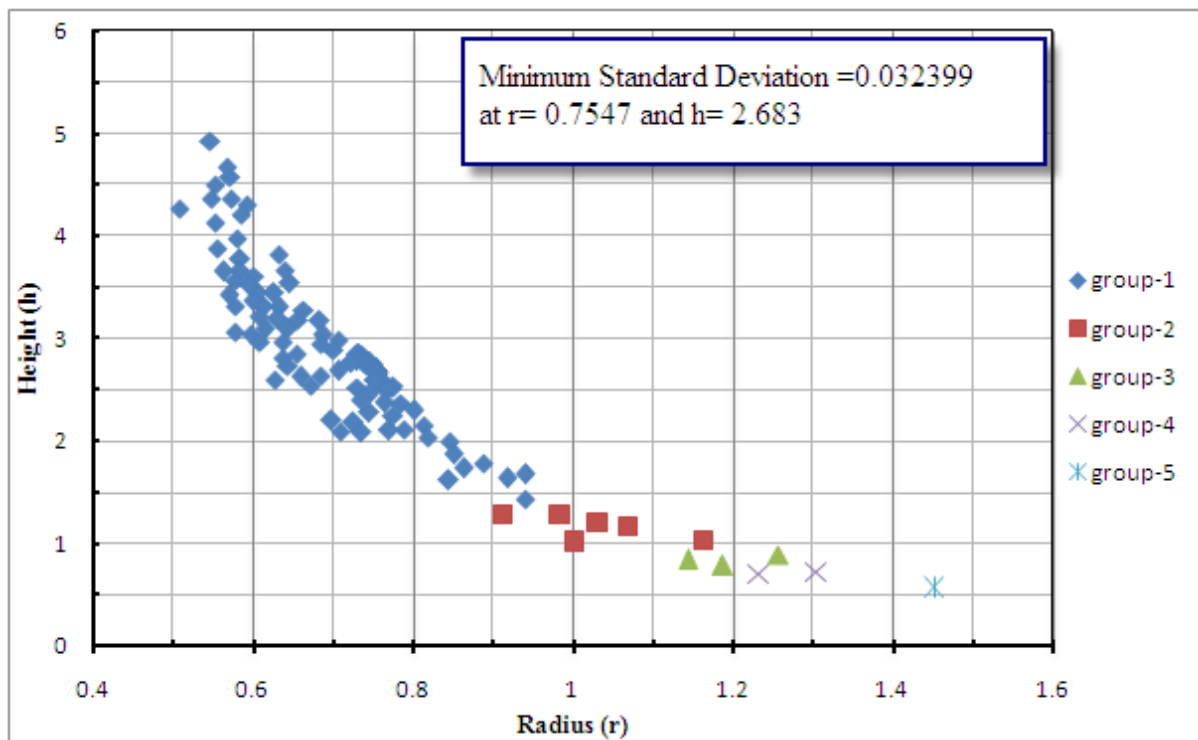


Figure 5-8: Distribution of robust design table of FMOGA –II 100-10 with 20% tolerance

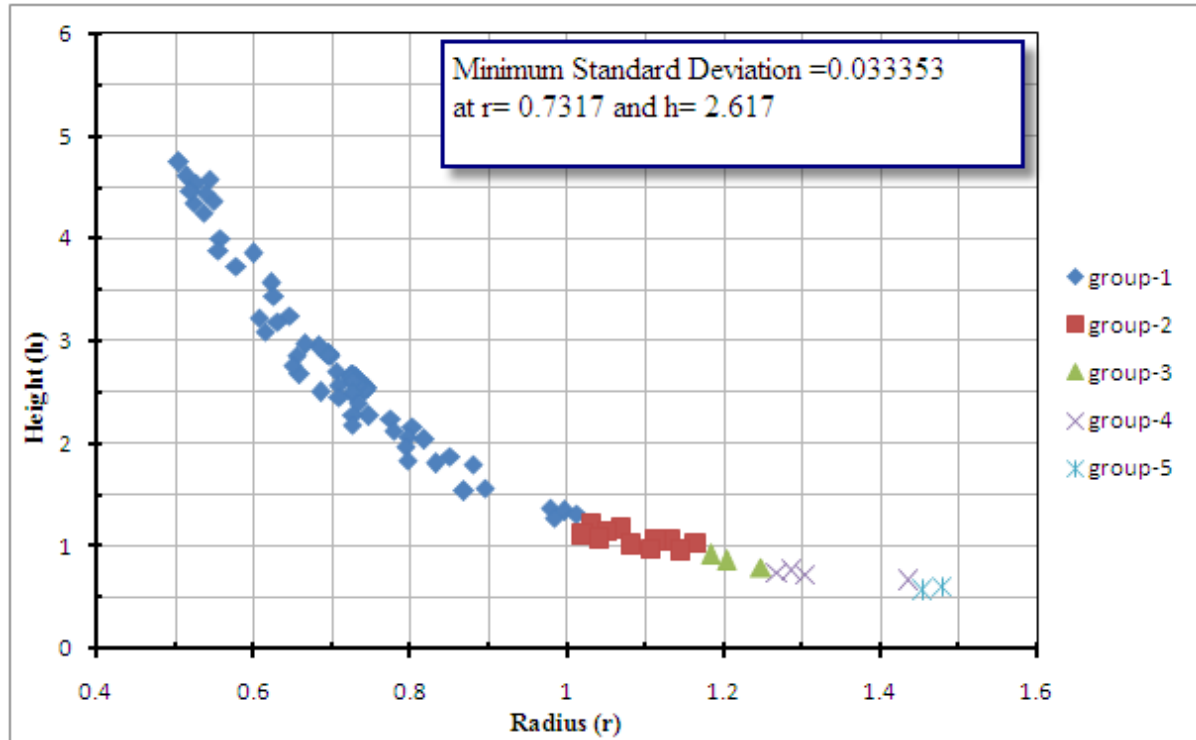


Figure 5-9: Distribution of robust design table of FMOGA –II 100-10 with 10% tolerance

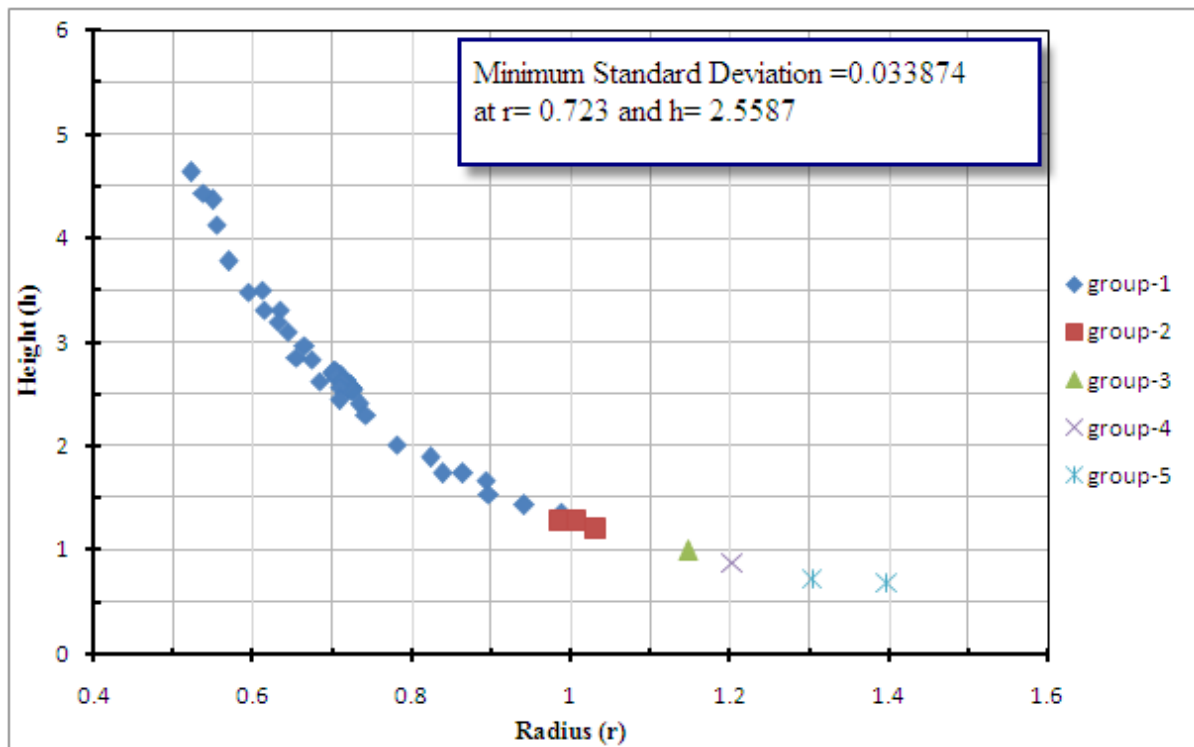


Figure 5-10: Distribution of robust design table of FMOGA –II 100-10 with 5% tolerance

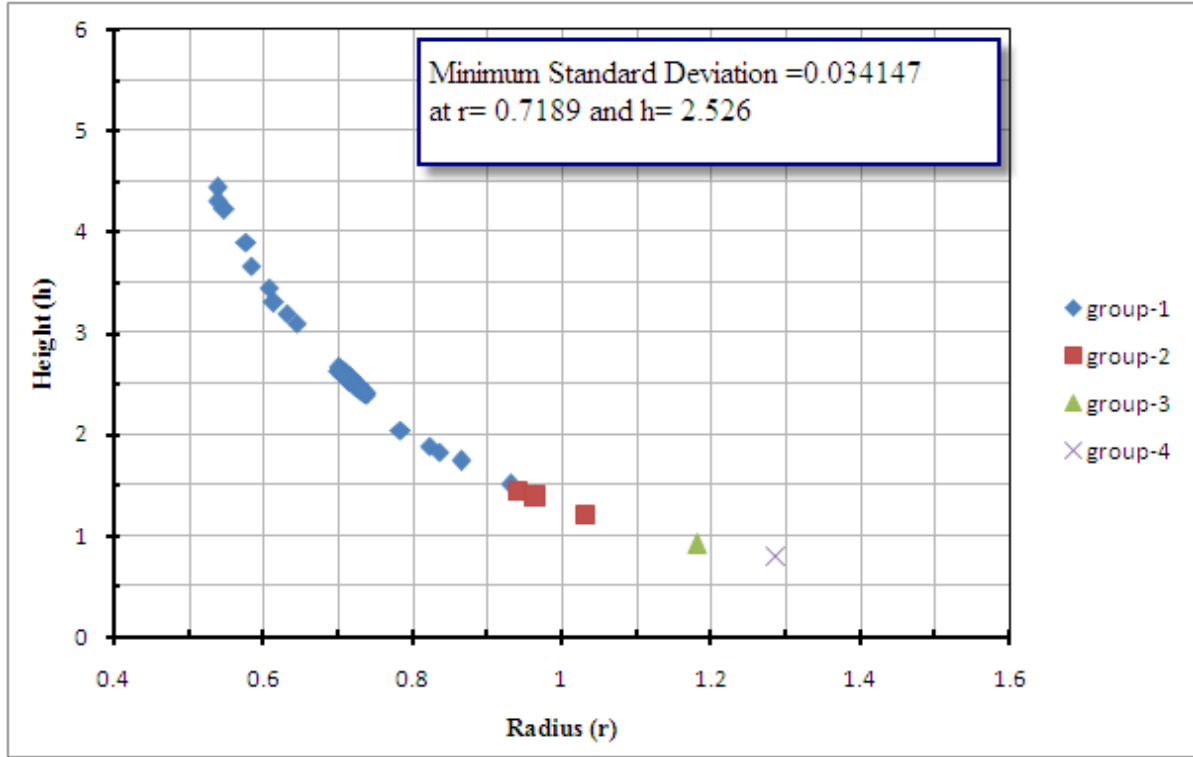


Figure 5-11: Distribution of robust design table of FMOGA –II 100-10 with 2.5% tolerance

5.3 Robust Optimisation of Lifting Arm Problem

Referring back to the problem of optimisation of the lifting arm (Chapter Four), the traditional optimisation of the lifting arm with the following objectives function:

$$\text{Minimizing } f_1 = C_{Y,t} = \frac{1}{Y_t} = \frac{1}{L (\cos \theta_{\min,t} - \cos \theta_{\max,t})} \quad (5.8)$$

$$\text{Minimizing } f_2 = C_{R,t} = \frac{1}{R_t} = \frac{L \sin \theta_{\max,t}}{d_{2,t} F \sin \beta_t} \quad (5.9)$$

The robust optimisation process of this problem consists of two approaches depending on the initial population of the robust optimisation.

5.3.1 Approach 1

The robust optimisation process in this approach consists of the following steps:

1. Perform the traditional optimisation (deterministic analysis) that uses the mean values of input parameters with the following values indicated in Table 5.3:

Table 5-3: Traditional optimisation of lifting arm problem parameters

<i>Input variables</i>	$0.1 \leq x_1 \leq 0.9$ and $0.1 \leq x_2 \leq 0.9$
<i>Initial Population</i>	64 Sobol
<i>Number of Generations</i>	1 FMOGA-II
<i>Number of Pareto solution</i>	24

2. Feeding the Pareto front solution from an initial multi-criteria optimisation step to the stage of robust optimisation as an initial population. The tolerance is equal to 3 standard deviations so with 1 mm as a tolerance on design variables the standard deviation will be equal to 0.000333. The aim is to minimise the standard deviation on the output or system response (objective functions).

Table 5-4: Robust optimisation of lifting arm problem parameters

<i>Input variables mean values</i>	$x_1 = 0.5$ and $x_2 = 0.5$
<i>Input variables standard deviation</i>	$x_1 = 0.000333$ and $x_2 = 0.000333$
<i>Initial Population</i>	24 from the first step
<i>Number of Generations</i>	10 FMOGA-II

3. Three different type of analysis have been studied regarding the tolerances on the objective function (by using the same technique as in the tank problem, see Section 5.2.1) with ($\pm 20\%$, $\pm 10\%$, and $\pm 5\%$), the model definition within the modeFRONTIER software is presented in Figure 5.12.

The functions that are used to give the approximation to the Pareto front line and tolerance lines are presented in Table 5.5, whilst the graphical representation of the Pareto front 20% and 5 % tolerance lines are presented in Figures 5.12 to 5.14.

Table 5-5: Approximation of Pareto front and constraint lines

Tolerances	Equation
Pareto Front line	$f(x) = a x^{-1}$
+20 % line	$f(x) = a x^{-1.034}$
-20 % line	$f(x) = a x^{-0.9737}$
+10 % line	$f(x) = a x^{-1.0166}$
-10 % line	$f(x) = a x^{-0.9866}$
+5 % line	$f(x) = a x^{-1.0085}$
-5 % line	$f(x) = a x^{-0.9739}$
a	0.0051

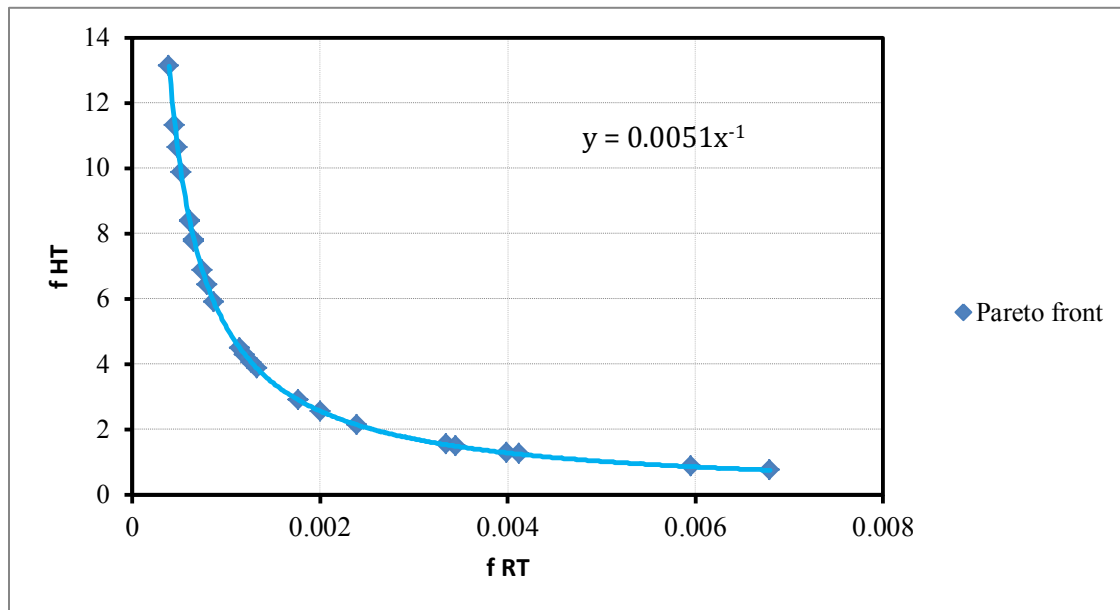


Figure 5-12: Representation of Pareto front approximation

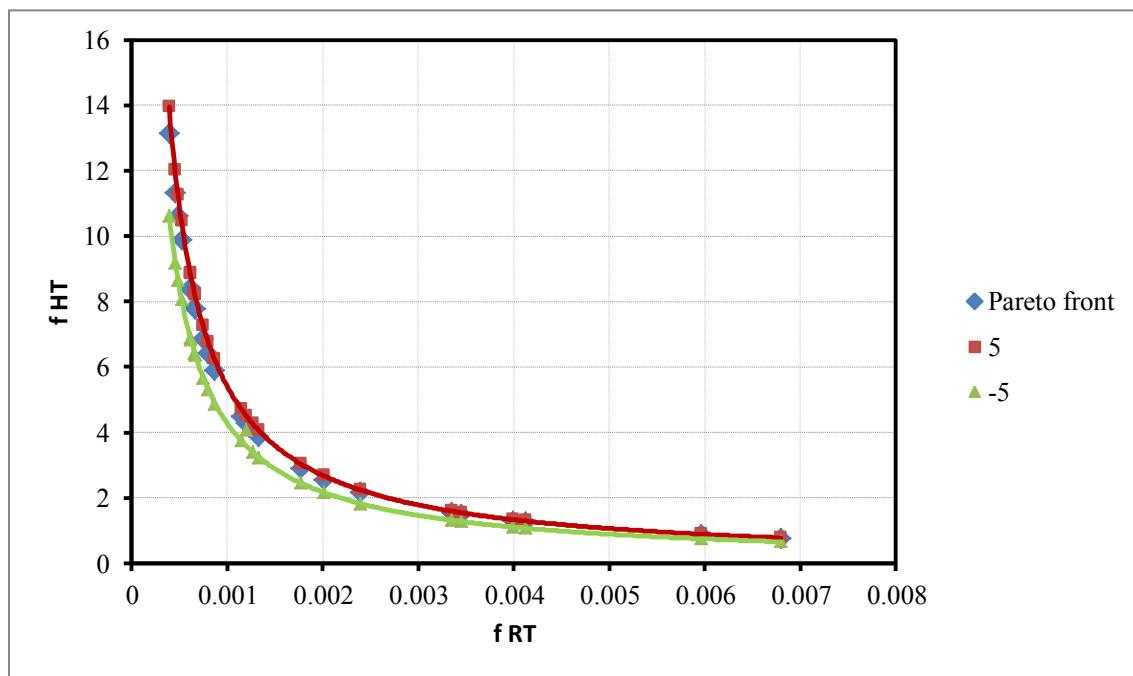


Figure 5-13: Pareto front with (+5%,-5%) constraint line

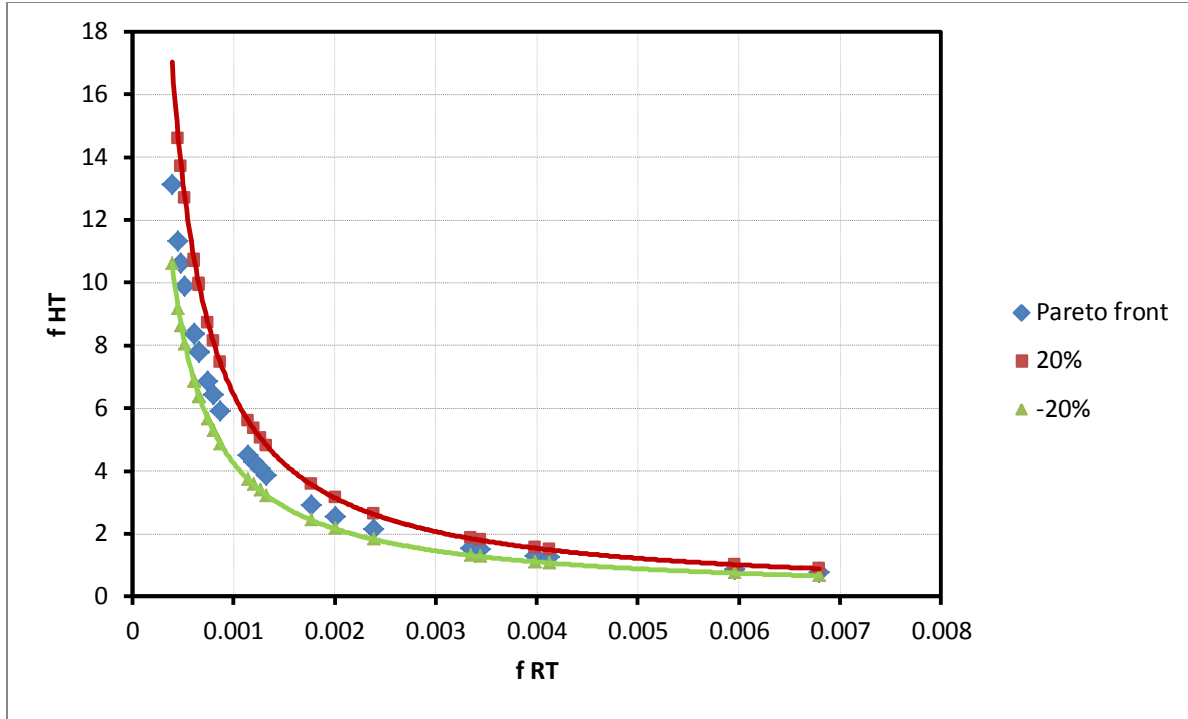


Figure 5-14: Pareto front with (+20%,-20%) constraint line

The distribution of Pareto robust designs in design and objective standard deviation space is presented in Figures 5.16 and 5.17. Table 5.6 indicates the result of this approach for cases 1, 2 and 3 with ($\pm 20\%$, $\pm 10\%$, $\pm 5\%$ tolerance band). It is clear that the increase of constraint will not have any effect on the final results and all cases show the same standard deviation of objective space.

5.3.2 Approach 2:

The main difference between this approach and approach 1 is that the initial population in this case is generated by using Sobol. In order to compare the results between this approach and approach 1, three different cases with three tolerance bands have been studied with the same parameters as in the previous approach. Figures 5.18 to 5.23 indicate the distribution of Pareto robust design in design and objective space. The results of this approach are indicated in (case 4, case 5 and case 6) in Table 5.6. It is evident that as the tolerance on the Pareto set is decreased the amount of scatter in objective standard deviation space is reduced. Also, the position of the

robust point moves in the design parameter space and is only coincident with the point found in cases 1, 2 and 3 for the tightest tolerance.

5.3.3 Approach 3:

To study the effect of initial population on the final results, two more cases have been investigated by increasing the initial population in case-4 from 24 to 50 and 100. Figures 5.24 to 5.27 show the distribution of the robust solution in design and objective standard deviation space whilst the results of this approach are indicated in Table 5.7. As can be seen in this table, the increase in the initial population 24, 50 and 100 will decrease the standard deviation, which means the quality of the solution will increase by increasing the initial population as a result of increasing the design space covered by the initial population.

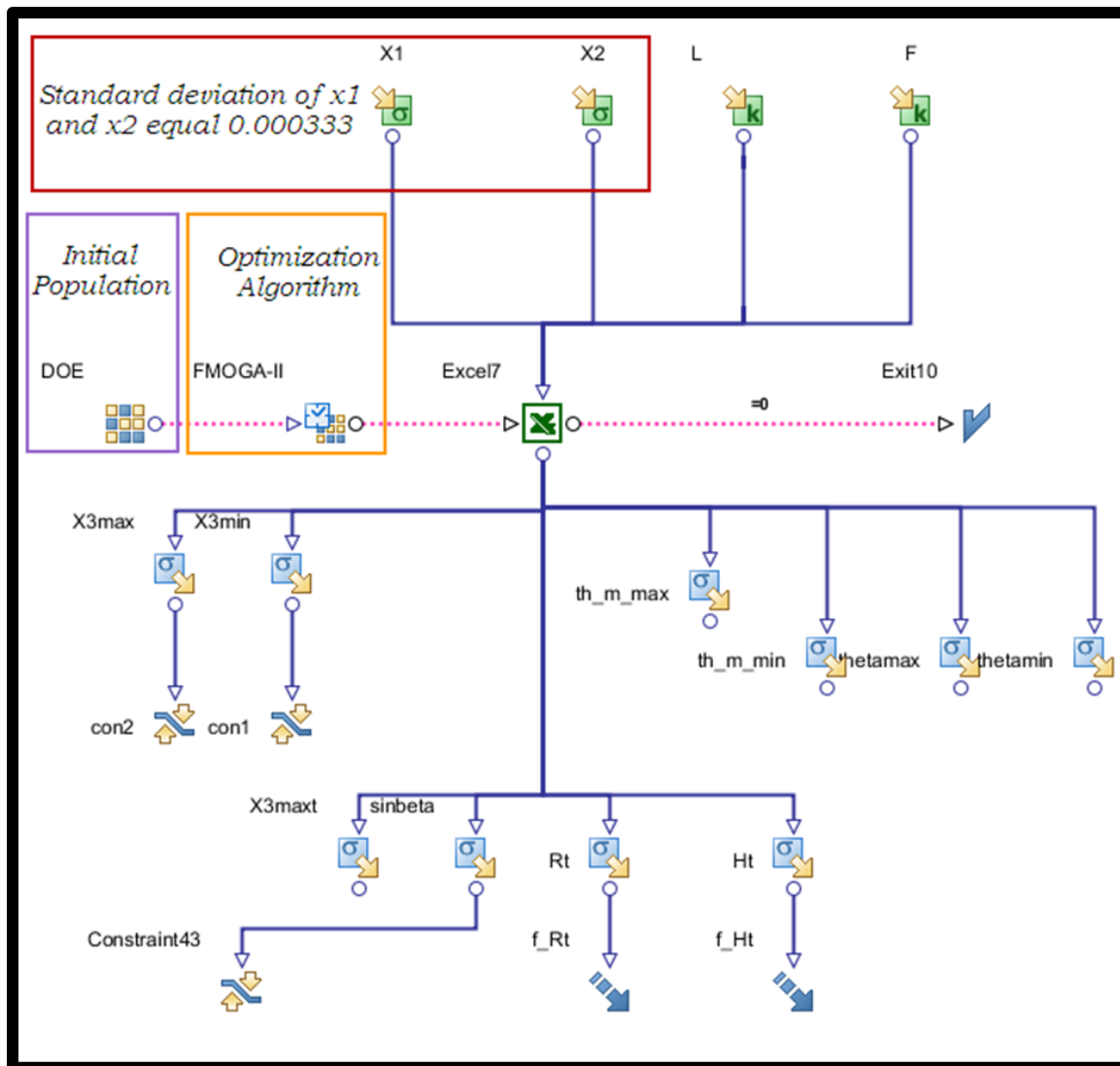


Figure 5-15: Robust optimisation of lifting arm problem in modeFRONTIER software

Table 5-6: Robust optimisation results for the first and second approach

First approach results					
Case no	In all the cases below the initial population = 24 from traditional optimisation And 10 generations with FMOGA-II Algorithm				
1	Robust with 20% tolerance-STD-0.000333	Total designs	96	Standard deviation on the first objectives	5.2321E-4
		Error designs	9		
		Non-feasible	0	Standard deviation on the second objectives	5.2321E-4
		Marked designs	1		
2	Robust with 10% tolerance-STD-0.000333	Total designs	96	Standard deviation on the first objectives	5.2321E-4
		Error designs	9		
		Non-feasible	0	Standard deviation on the second objectives	5.2321E-4
		Marked designs	1		
3	Robust with 5% tolerance-STD-0.000333	Total designs	96	Standard deviation on the first objectives	5.2321E-4
		Error designs	9		
		Non-feasible	0	Standard deviation on the second objectives	5.2321E-4
		Marked designs	1		
Second approach results					
	In all the cases below the initial population = 24 from Sobol And 10 generations with FMOGA-II Algorithm				
4	Robust with 20% tolerance-STD-0.000333	Total designs	144	Standard deviation on the first objectives	5.1639E-4
		Error designs	18		
		Non-feasible	0	Standard deviation on the second objectives	5.1640E-4
		Marked designs	1		
5	Robust with 10% tolerance-STD-0.000333	Total designs	120	Standard deviation on the first objectives	5.1942E-4
		Error designs	22		
		Non-feasible	0	Standard deviation on the second objectives	5.1938E-4
		Marked designs	1		
6	Robust with 5% tolerance-STD-0.000333	Total designs	72	Standard deviation on the first objectives	5.1719E-4
		Error designs	15		
		Non-feasible	0	Standard deviation on the second objectives	5.1723E-4
		Marked designs	1		

Table 5-7: Robust optimisation results for the third approach

Third approach results					
Case no	Initial Population = 24 Sobol				
4	Robust with 20% tolerance-STD-0.000333	Total designs	144	Standard deviation on the first objectives	5.1639E-4
		Error designs	18		
		Non-feasible	0	Standard deviation on the second objectives	5.1640E-4
		Marked designs	1		
7	Initial Population = 50 Sobol				
	Robust with 20% tolerance-STD-0.000333	Total designs	250	Standard deviation on the first objectives	5.1427E-4
		Error designs	80		
		Non-feasible	0	Standard deviation on the second objectives	5.1425E-4
Marked designs		1			
8	Initial Population = 100 Sobol				
	Robust with 20% tolerance-STD-0.000333	Total designs	500	Standard deviation on the first objectives	4.7930E-4
		Error designs	107		
		Non-feasible	0	Standard deviation on the second objectives	4.7933E-4
Marked designs		1			

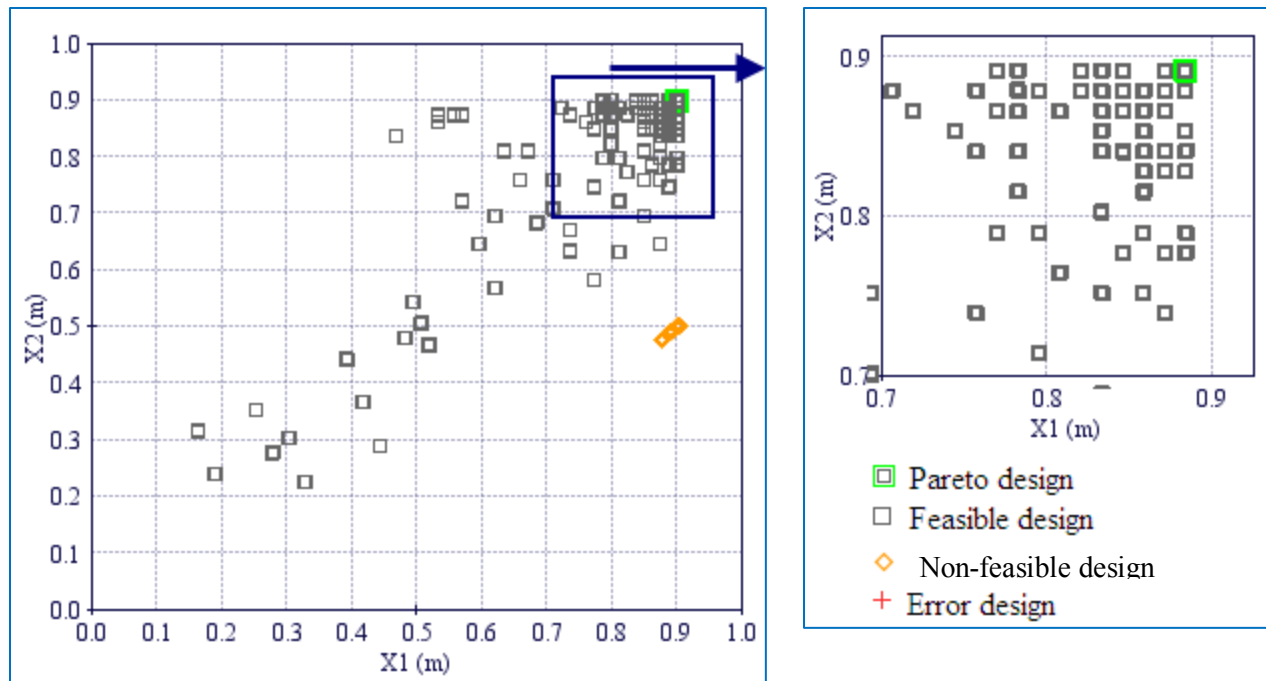


Figure 5-16: Distribution of Pareto robust design in design space for cases (1, 2 and 3)

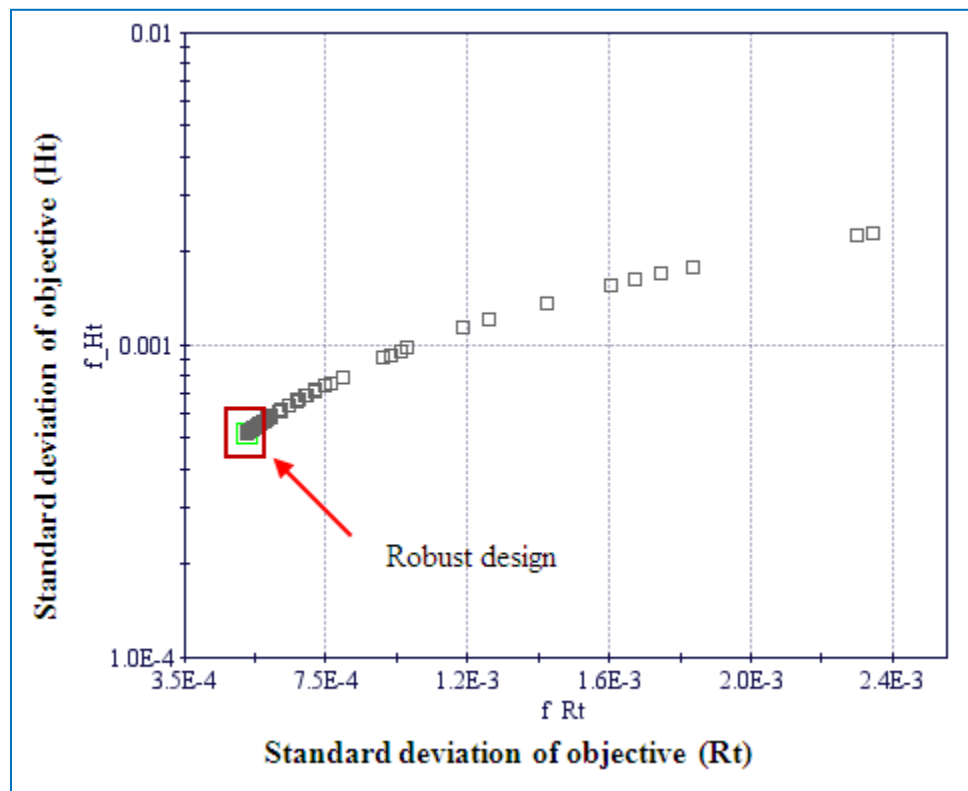


Figure 5-17: Relation between standard deviation of (objective1, objective2) for cases (1, 2 and 3)

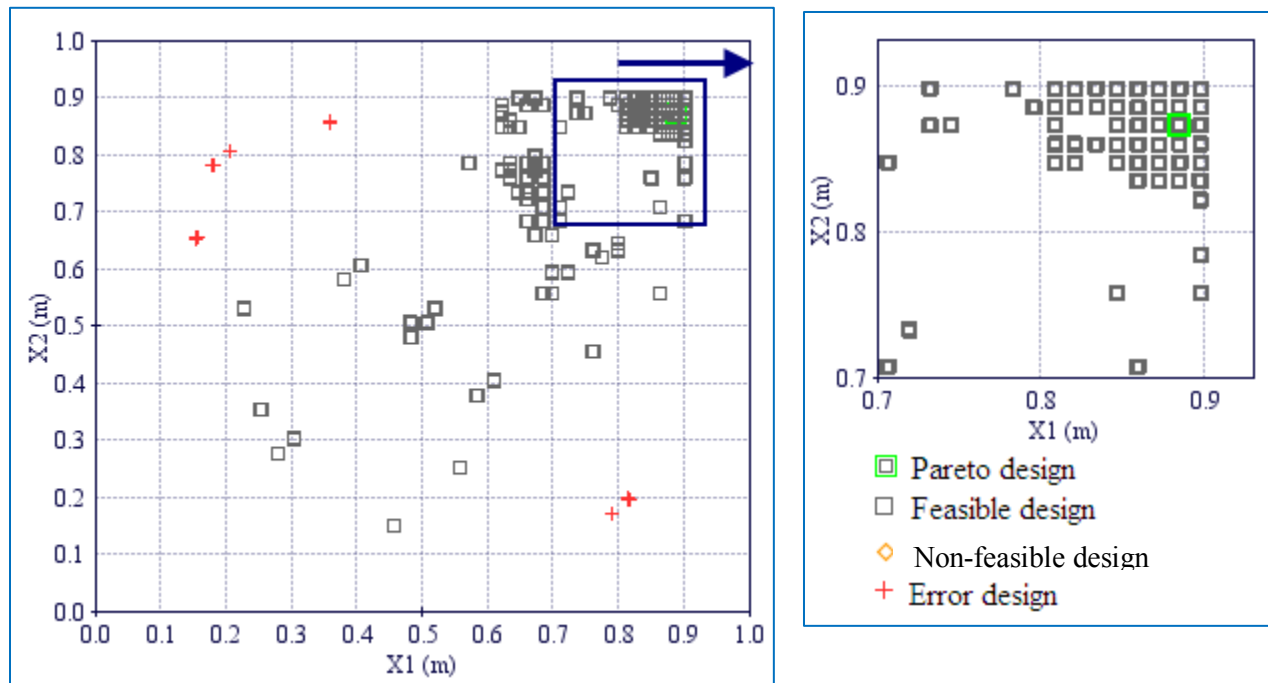


Figure 5-18: Distribution of Pareto robust design in design space for case (4)

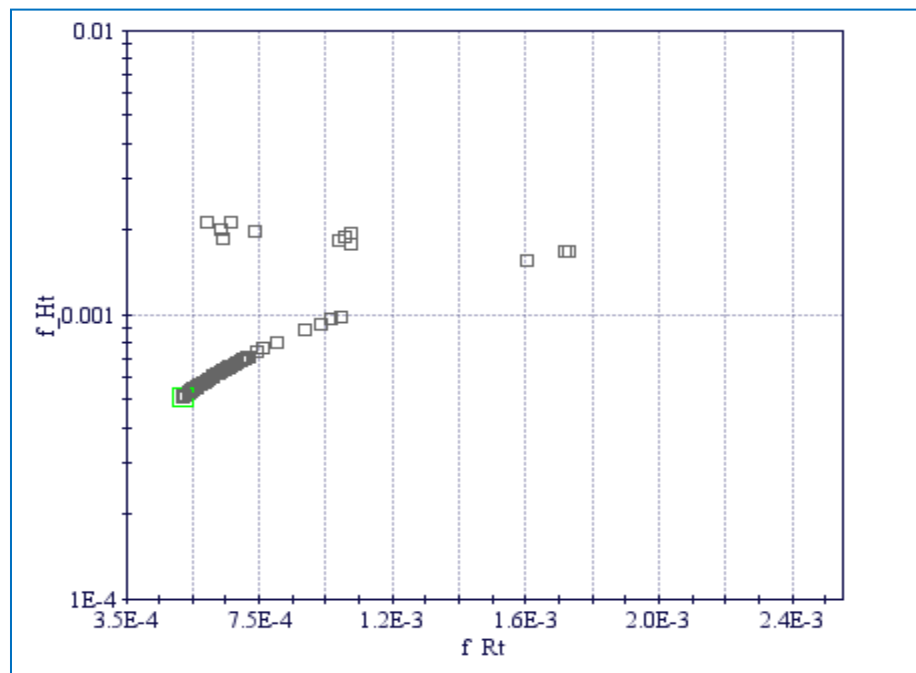


Figure 5-19: Relation between standard deviation of (objective1, objective2) for case (4)

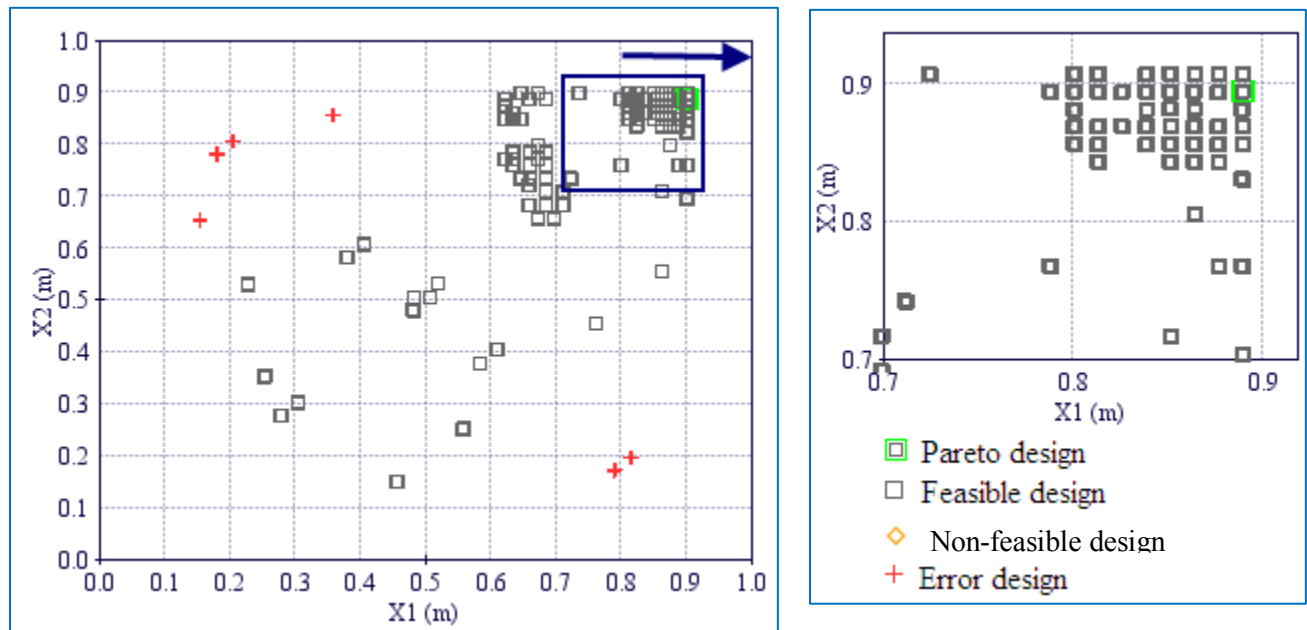


Figure 5-20: Distribution of Pareto robust design in design space for case (5)

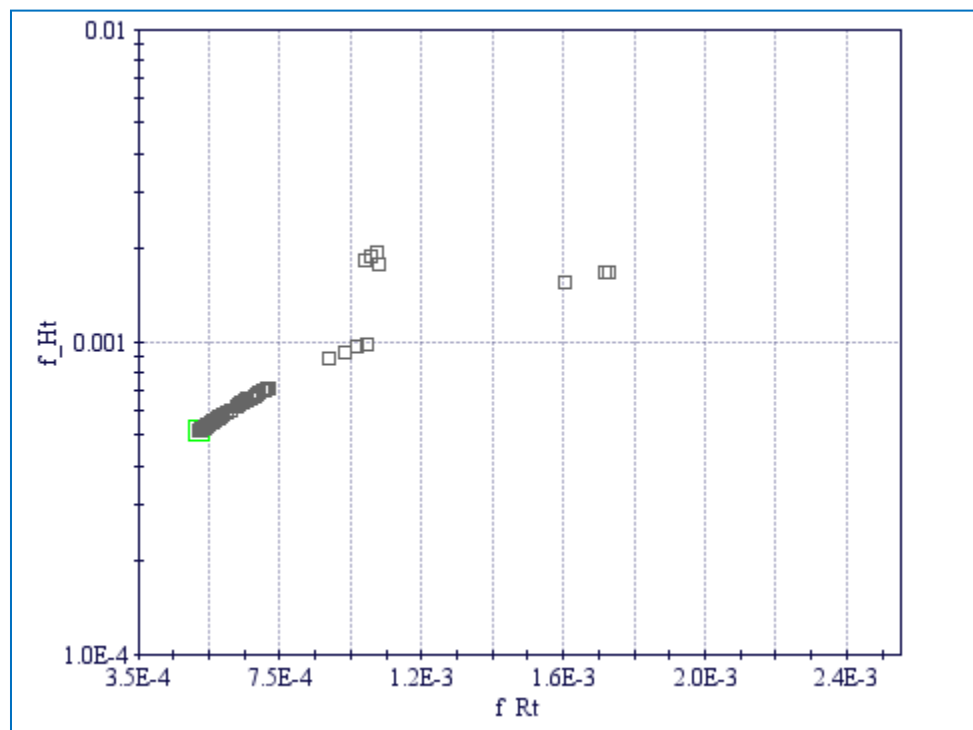


Figure 5-21: Relation between standard deviation of (objective1, objective2) for case (5)

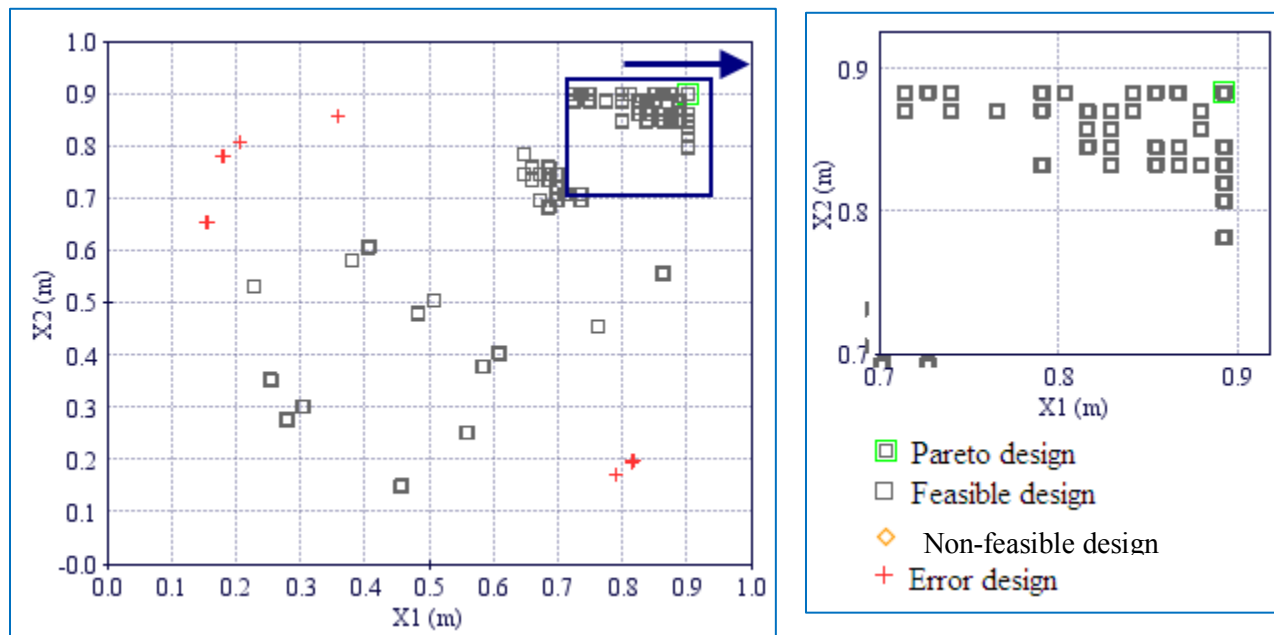


Figure 5-22: Distribution of Pareto robust design in design space for case (6)

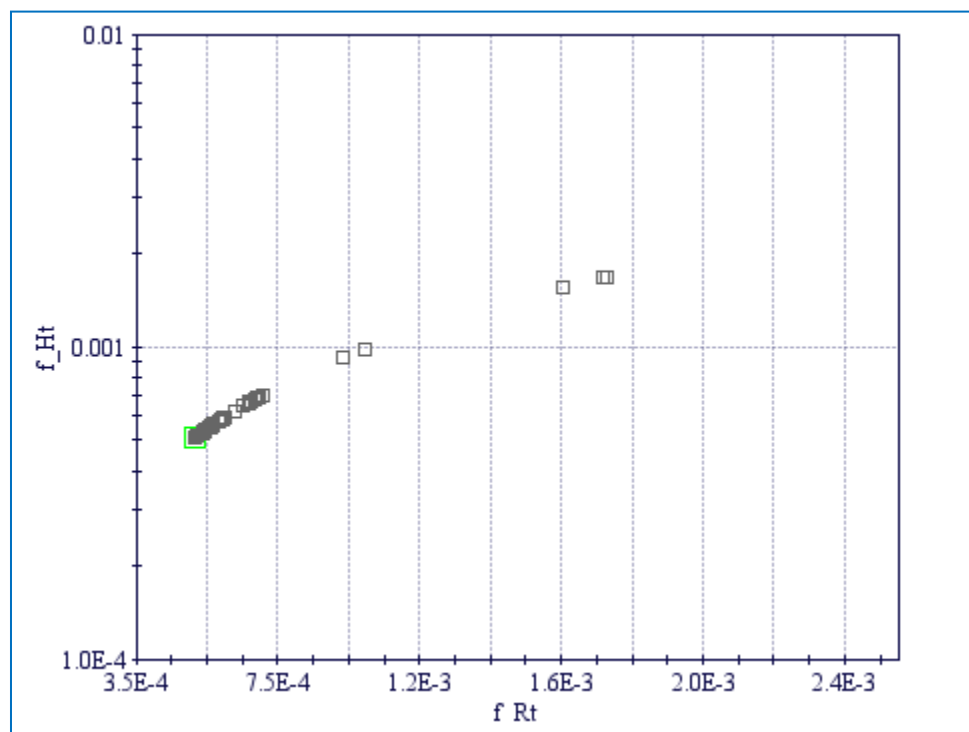


Figure 5-23: Relation between standard deviation of (objective1, objective2) for case (6)

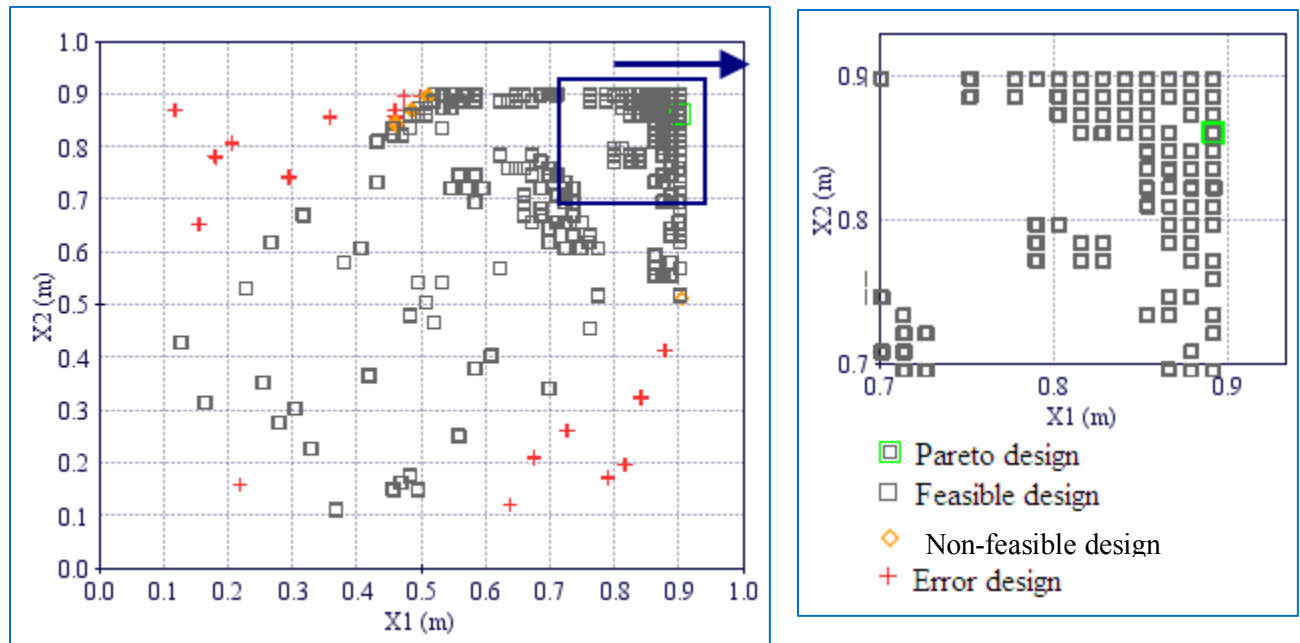


Figure 5-24: Distribution of Pareto robust design in design space for case (7)

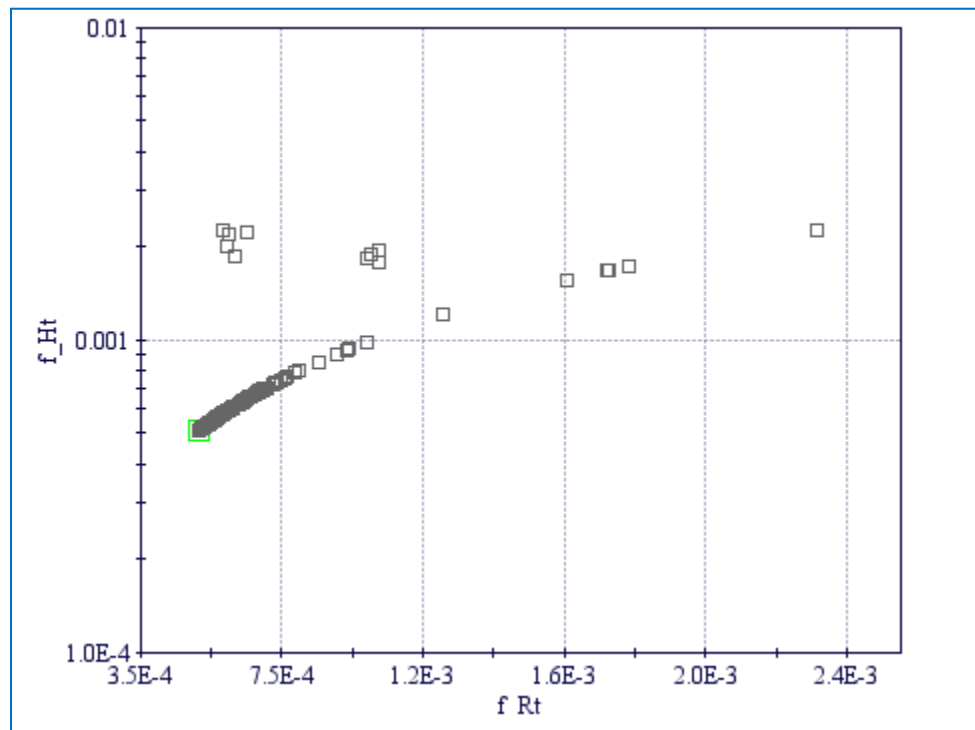


Figure 5-25: Relation between standard deviation of (objective1, objective2) for case (7)

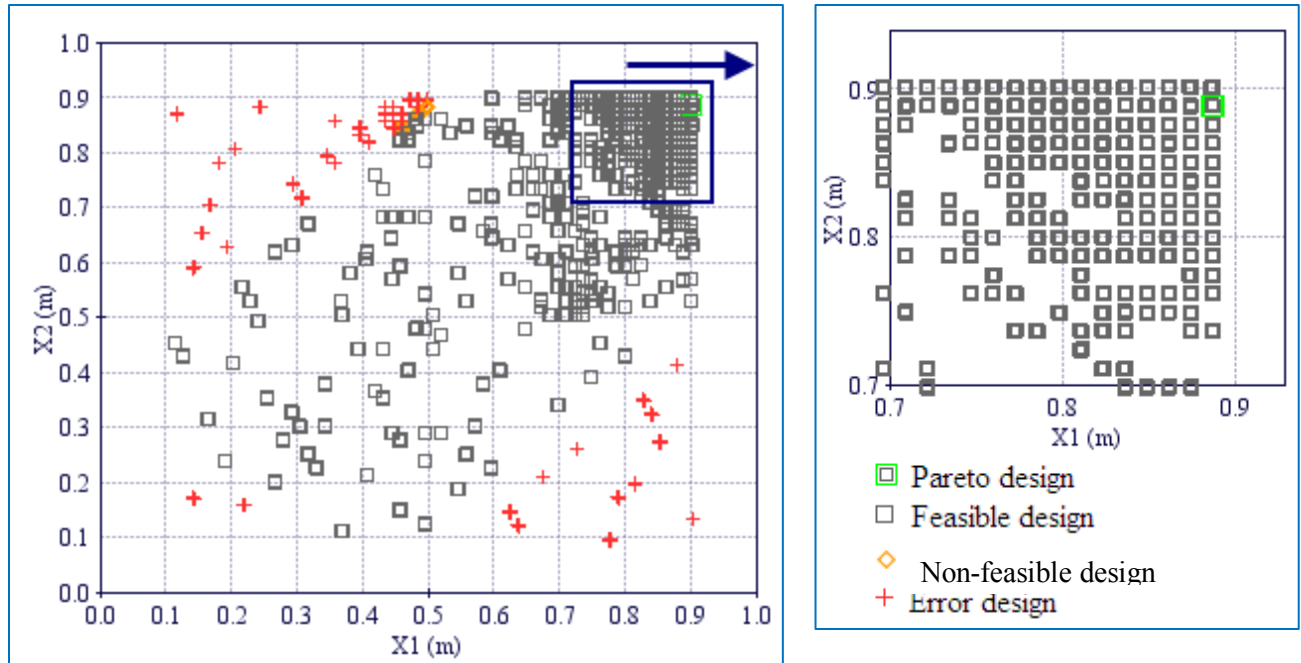


Figure 5-26: Distribution of Pareto robust design in design space for case (8)

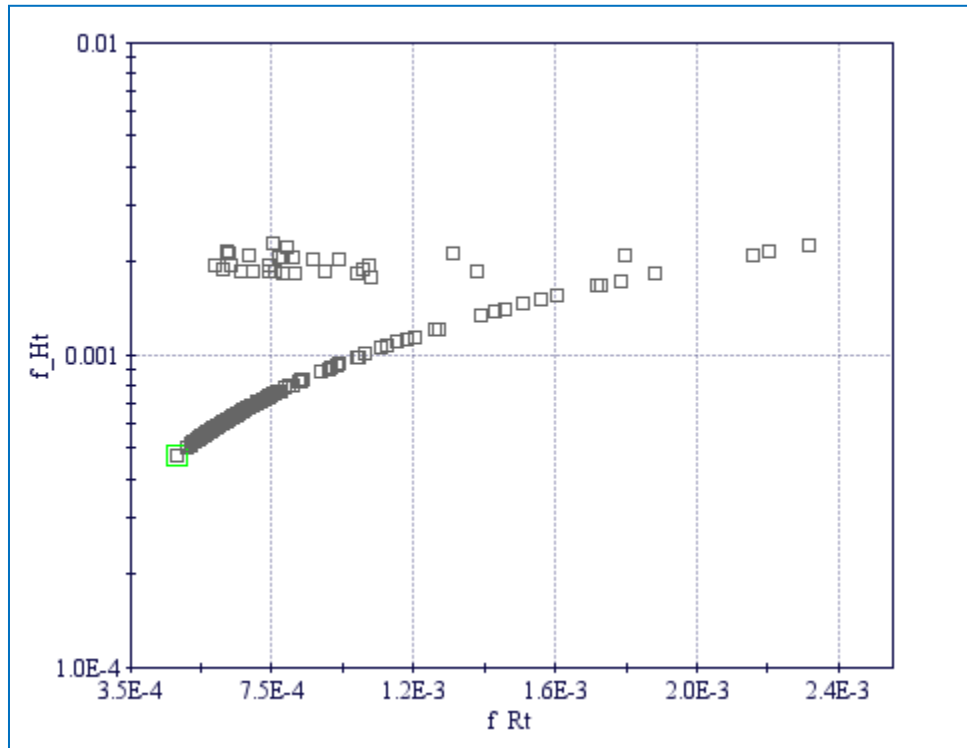


Figure 5-27: Relation between standard deviation of (objective1, objective2) for case (8)

5.4 Conclusion

- In the tank problem, reducing the tolerance on the objective function (i.e. the volume of the tank) increased the standard deviation of the robust solution due to reducing the search space.
- Using a uniformly distributed initial population for the lifting arm problem resulted in greater scattering of points both in design space and in objective standard deviation space. Once the tolerance on the Pareto set was reduced, the amount of scatter (in objective standard deviation space) was reduced and the robust solution converged towards that found for approach 1.
- Using more points in the initial Sobol set was found to be inefficient, producing more non-feasible points, more points outside the tolerance band on the Pareto front and only converging to the previously identified robust solution with large initial populations.
- The robust design in the second problem is sited on the maximum value of design variables (x_1 and x_2). This is the expected result since an absolute tolerance on these two design variables was used. This gives confidence that the method is working correctly.

Chapter 6

Finite Element Method

Chapter Six: Finite Element Method

6.1 Fundamentals of the Finite Element Method

The Finite Element Method (FEM) is used in mathematics for identifying an approximate solution to boundary value problems for differential equations. The finite element method, today, is widely used in almost all fields of science and engineering, such as, structural engineering, structural dynamics, aerodynamics, aeroelasticity, fluid flow, thermodynamics, foundation engineering, soil mechanics, geotechnical engineering, pile foundation, machine foundation, bearing, lubrication, nuclear containment systems, fluid soil structure interaction, textile engineering, electrical technology and cable systems etc.

The basic idea of finite element is to divide the whole geometry into a finite number of small elements “The domain of the problem is viewed as a collection of nonintersecting simple subdomains, called finite elements. The subdivision of a domain into elements is termed finite element discretization. The collection of elements is called the finite element mesh of the domain” Reddy, 1993. The assumed advantage of dividing the whole geometry into small elements with a simple shape is that within an individual element, the change in the solution variable can be approximated by a relatively simple interpolation polynomial whereas in the whole structure a much more complex function would be required. This assumption is only valid if, in regions where the solution variable changes rapidly, the individual elements are sufficiently small.

6.1.1 Procedure of FEA

In any finite element analysis the basic steps consist of the following:

1. *Discretization of the structure or solution region into a number of smaller parts, or regions called elements.*
2. *Propose a shape function (interpolation polynomial) to represent the physical behaviour of an element.*
3. *Develop equations for an element.*
4. *Assembly the connected elements to create one large set of equations.*
5. *Application of external loads and boundary conditions.*
6. *Solution for nodal displacements.*
7. *Determine the stresses, strains, reactions etc. from the nodal displacements..*

6.1.2 Abaqus Basics

Abaqus is an engineering simulation programs based on the finite element method. Abaqus/CAE provides a pre-processing and post-processing environment for the analysis of the model. Due to its capability to solve relatively simple linear problems to the most challenging nonlinear simulations it is used in a wide range of industries like aerospace, automotive, structural analysis etc., and also is extensively used in academic and research institution.

There are three distinct stages in a full Abaqus analysis:

1. Pre-Processing
2. Simulation
3. Post-Processing

These are connected by files as shown in Figure 6.1:

1. Pre-processing (Abaqus / CAE)

In this stage, a physical model is defined and an Abaqus input file created. This is usually carried out using the Abaqus/CAE graphical user interface (GUI) or another pre-processor, although it is possible to create an Abaqus input file (Job.inp) for a simple analysis by using a text editor.

2. Simulation (Abaqus / Standard)

The model input file (Job.inp) is submitted to the solver. The Job times in the solver are dependent on different parameters such as number of elements and element type and in particular whether the material properties or boundary conditions (load and constraint) introduce any non-linearity into the analysis.

3. Post-processing (Abaqus / CAE)

Results evaluation is generally interactive and this stage comes after the completion of simulation and the final calculations of displacements, stresses and other variables. These interactive sessions are performed using the visualisation module of Abaqus / CAE. This module accesses the binary output file and displays the results in varying forms, including colour contour plots, animations, deformed shape plots and X–Y plots.

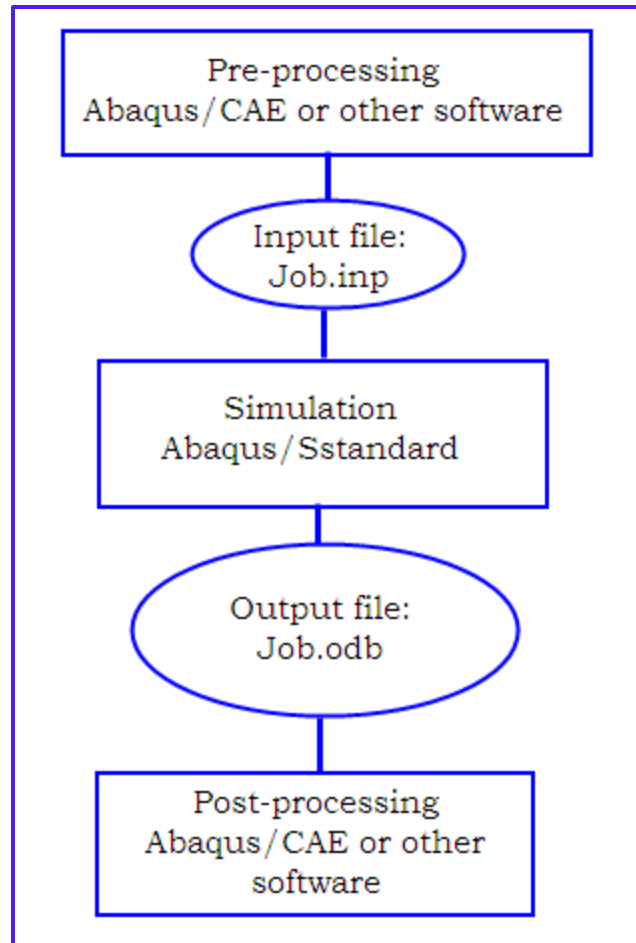


Figure 6-1: Abaqus stages of analysis Abaqus (2008)

6.2 Modelling the Welded Joint

6.2.1 Welded joint parameters

Figure 6.2 and Table 6.1 indicate the geometrical parameters of the welded joint. It is clear that the geometry of the joint is symmetrical about the y-axis and due to this symmetry the entire joint need not be analysed. One half of the model can be used. The boundary conditions of the welded joint are indicated in the Figure 6.4. These boundary conditions reduce the complexity of the joint and the time required to perform the analysis. Additionally, the geometrical parameters will be reduced from (11-full model) to (7 half model).

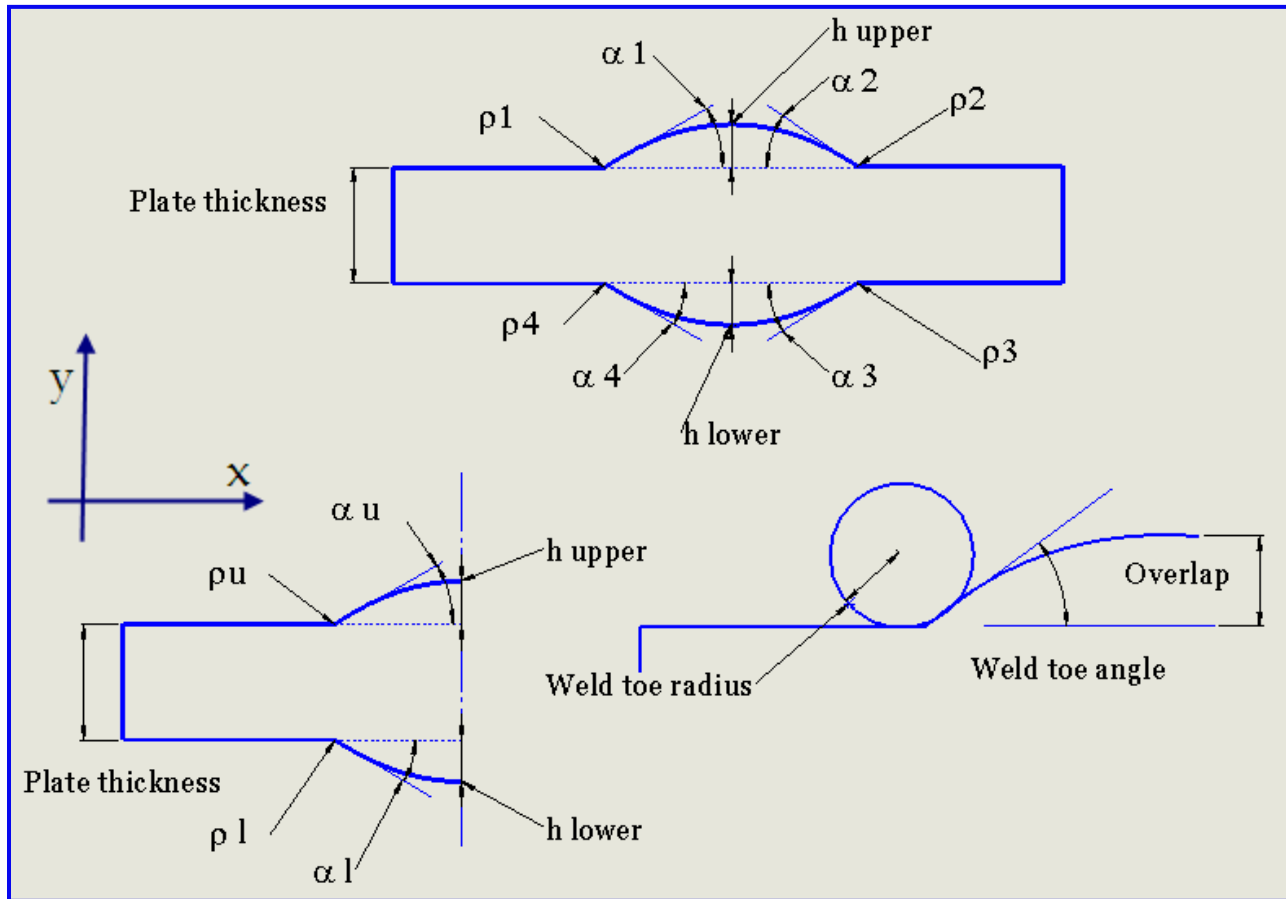


Figure 6-2: Geometry of butt welded joint

Table 6-1: Geometrical parameters of butt welded joint.

S. No	Symbol	Parameters	Units
1	t	Plate thickness	mm
2	h_u	Upper reinforcement	mm
3	h_l	Lower reinforcement	mm
4	ρ_1	Weld toe radius top-left	mm
5	ρ_2	Weld toe radius top-right	mm
6	ρ_3	Weld toe radius bottom-left	mm
7	ρ_4	Weld toe radius bottom-right	mm
8	α_1	Weld toe angle top-left	Degree
9	α_2	Weld toe angle top-right	Degree
10	α_3	Weld toe angle bottom-left	Degree
11	α_4	Weld toe angle bottom-right	Degree

Geometrical parameters after symmetry condition			
1	t	Plate thickness	mm
2	h _u	Upper reinforcement	mm
3	h _l	Lower reinforcement	mm
4	ρ_u	Weld toe radius top	mm
5	ρ_l	Weld toe radius bottom	mm
6	α_u	Weld toe angle top	Degree
7	α_l	Weld toe angle bottom	Degree

6.2.2 Material properties

The following material properties of steel S355JR are indicated in Table 6.2.

Table 6-2: Material properties of steel S355JR

Property	Symbol	Unit	Value
Young's Modulus	E	GPa	207
Poisson's ratio	ν	-	0.3
Tensile Yield strength	σ_y	MPa	370

6.2.3 Element Type

One, two, three-dimensional, and axisymmetric solid elements in Abaqus are named as follows (ABAQUS 2010b):

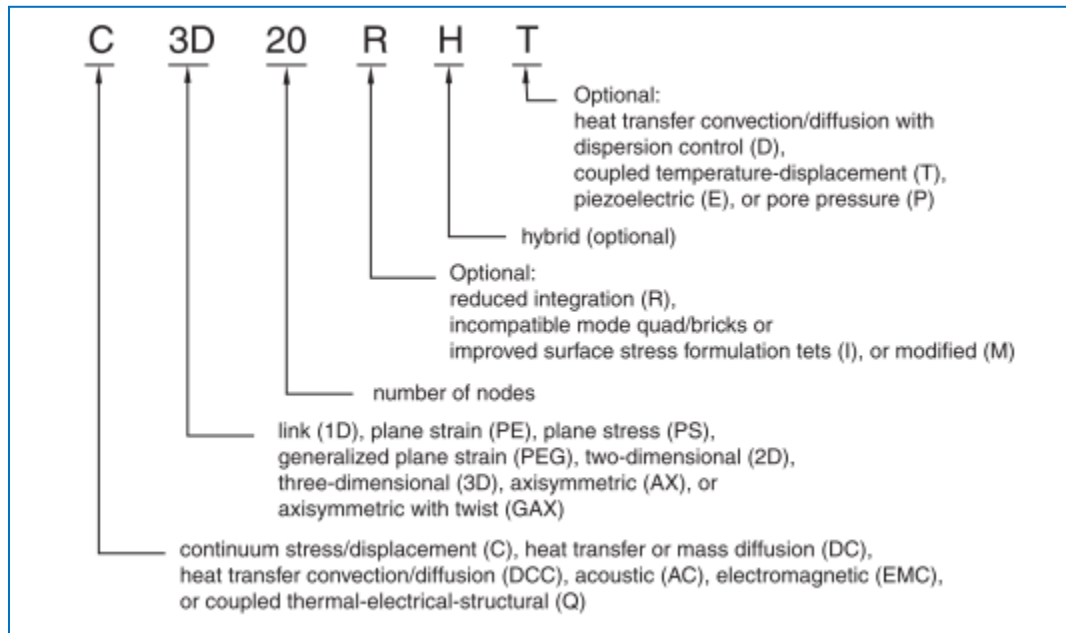


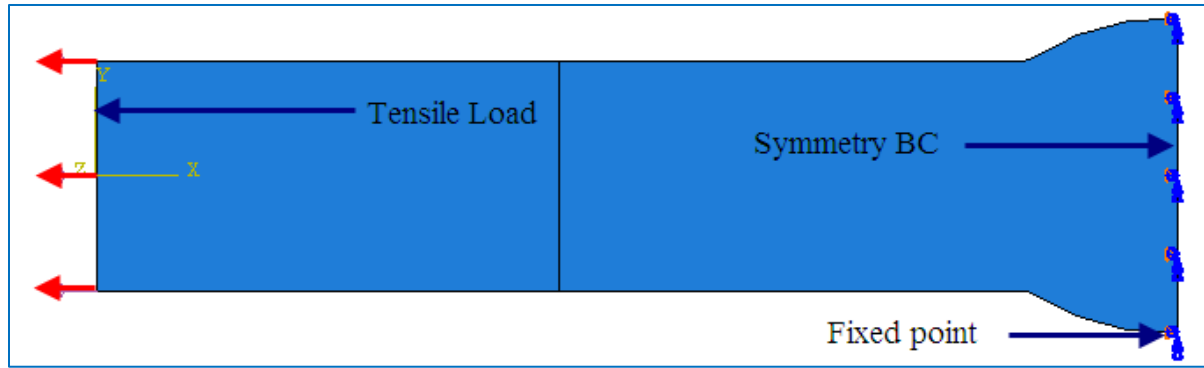
Figure 6-3: Abaqus element naming convention

For example, CAX4R is a 4-node, reduced-integration, axisymmetric continuum stress/displacement element. In the present study, the welded joint is meshed using the CPS4R bilinear, reduced integration element. This element has been selected rather than a higher order, full integration element in order to arrive at a solution more rapidly. Whilst a higher order element is often considered to be more efficient (i.e. requires fewer elements for the same accuracy of results), in the present study, in order to maintain an equivalent mesh density over a wide range of weld geometries, a fine mesh has to be used. The problems of spurious deformation modes due to reduced integration will not occur in the weld geometry analyses since the mesh density is so high.

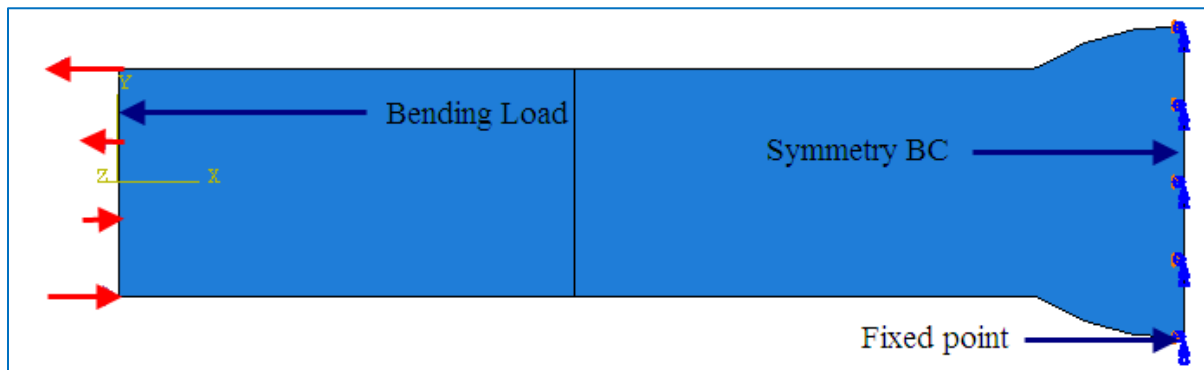
6.2.4 Load and Boundary Condition

In the numerical model, it is observed that the joint geometry and loading is symmetrical about the global Y axis. Hence, only one half of the joint is modelled. The main purpose of symmetric conditions is to reduce simulation time, but it also simplifies the application of constraints (on the plane of symmetry there can be no displacement perpendicular to the plane of symmetry). To prevent rigid body motion, an additional constraint is required in the y direction. This could be placed at any point in the model. In this case it is also placed at a point on the plane of symmetry.

The load and boundary condition of the joint are shown in Figure 6.4. The joint is subjected to two types of load: tensile load as shown in Fig. 6.4: (a) and bending load in Fig. 6.4: (b).



a)



b)

Figure 6-4: Welded joint under a) tensile load; and b) bending load

6.2.5 Model development

The accuracy of the results in Abaqus / CAE depends on the number of elements in each model and element type. The higher the number of elements in the model, the more accurate the analysis will be. However, with the increase in elements comes an increase in computational cost.

For the welded joint model, a convergence test was conducted to determine the least number of elements necessary for the tensile and bending stress results to converge to an acceptable degree. The results of this study are presented in Figure 6.5. Figures 6.6 and 6.7 show the mesh distribution and a contour plot of a joint under tensile and bending loads with the same geometry. These and similar joints will be used during the optimisation study.

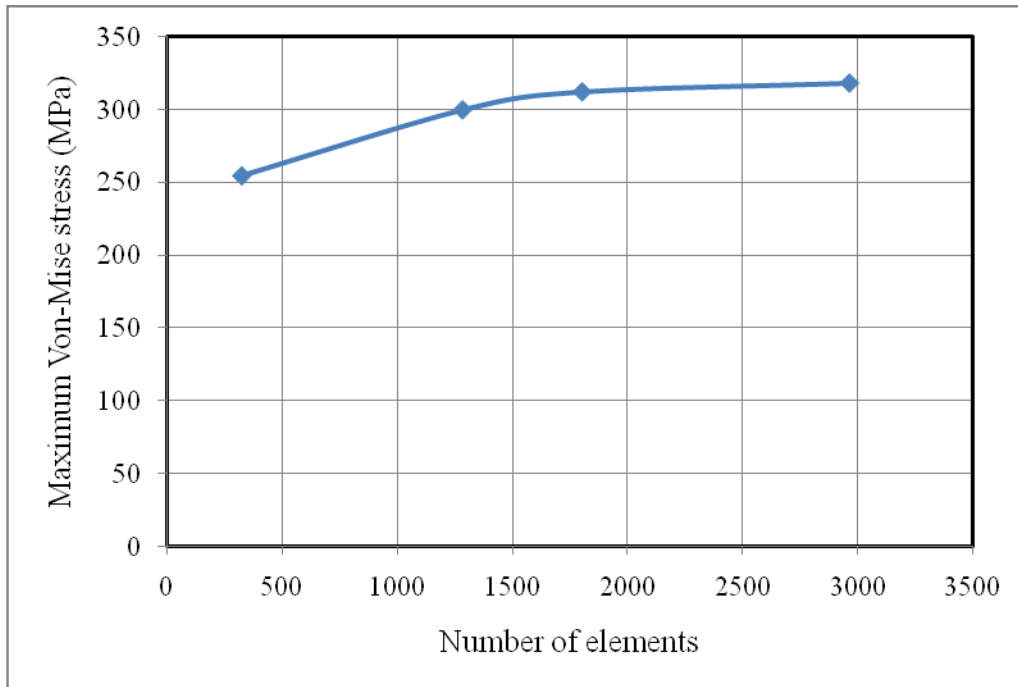


Figure 6-5: Mesh convergence study of welded joint under tensile loading

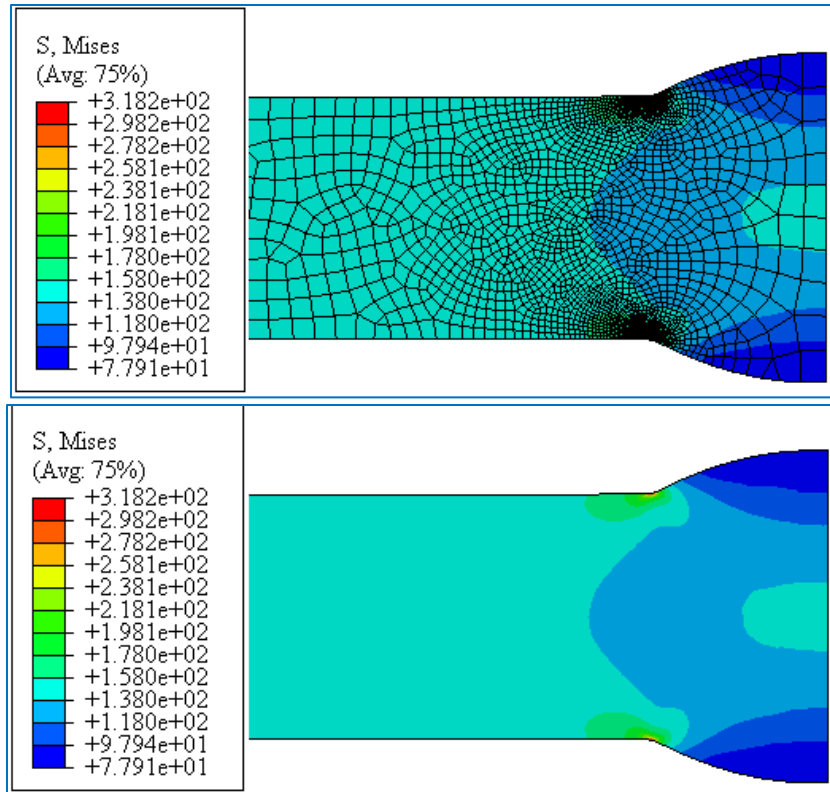


Figure 6-6: Mesh distribution and stress contour plot of joint under tensile loading

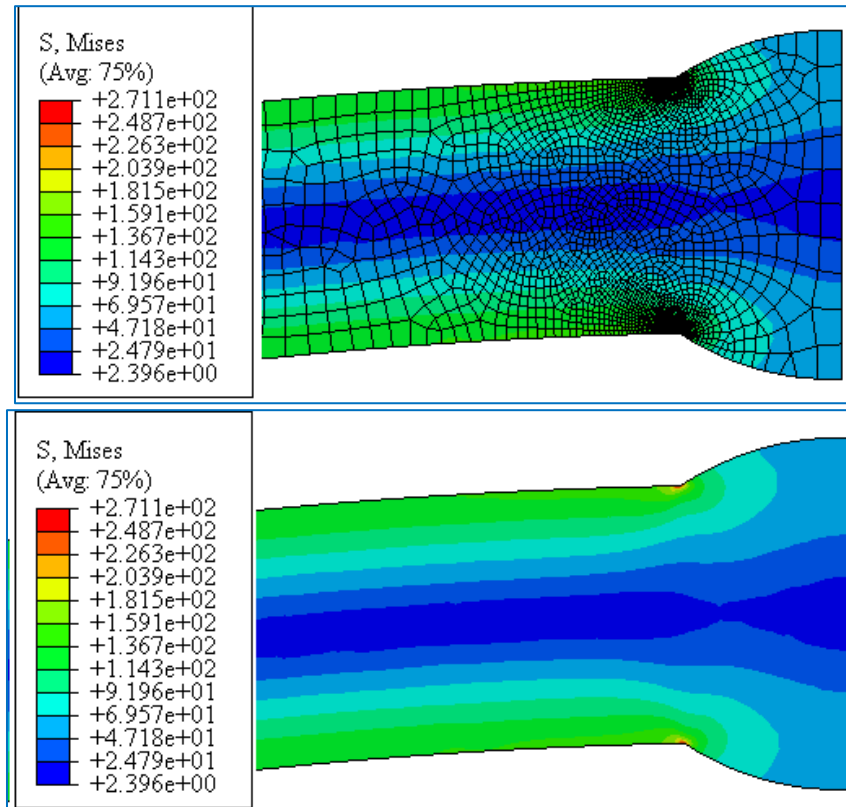


Figure 6-7: Mesh distribution and stress contour plot of joint under bending loading

6.3 Python Model Development

6.3.1 Python Programming language

Python is an object-oriented programming language that is widely used in the software industry. It is the standard programming language for Abaqus products and is used in several ways:

- The Abaqus environment file (abaqus_v6.env) uses Python statements.
- The parameter definitions in the Abaqus input file
- Abaqus/CAE records its commands as a Python script in the replay (Job.rpy) file.
- To execute Abaqus/CAE tasks directly using a Python script.
- To access the output database (Job .odb) using a Python script.

6.3.2 ABAQUS Scripting Interface

As illustrated in Figure 6.8, the Abaqus/CAE GUI uses the Python scripting language to produce the input file for the Abaqus solver. As described in Abaqus (2010a), it is therefore possible to use Python to communicate directly with the Abaqus/CAE kernel and generate input files without having to work through the GUI.

The Python communication feature of Abaqus/CAE is further enhanced by the ability to access the Python script that the GUI passes to the kernel for a given model. This script can be extracted, modified using standard Python commands and passed directly to the kernel. This feature makes it straightforward to generate parameter based models to investigate a range of similar models. This feature has been used extensively in the work described in this thesis.

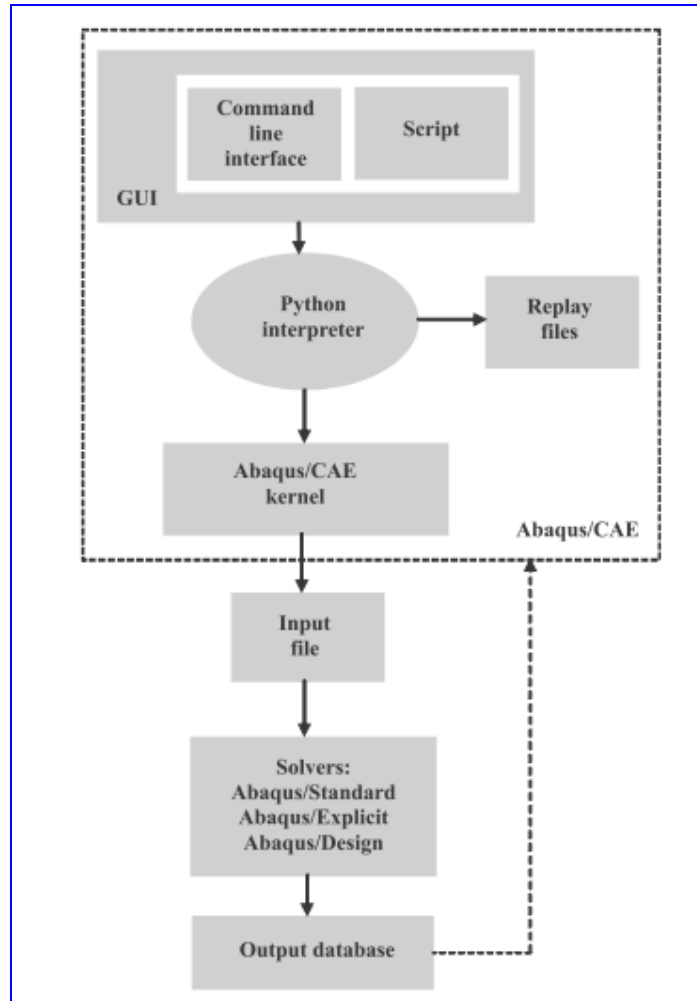


Figure 6-8: Abaqus scripting flow chart (Abaqus 2010a)

- **Creating Script**

Different methods can be used to create Python script. These methods are:

1. **By using Text editor:** The first method for creating script is by using a standard text editor like Text Pad, Notepad++, or using the Abaqus Python Development Environment (PDE). The Abaqus PDE application is developed for creating, modifying, testing and fixing Python scripts. Using the PDE or code is up to the choice of users using Python Syntax highlighting.
2. **Recording a Macro:** It is possible to register and record a sequence of command actions by using the 'Record Macro' button in the GUI. Once the 'Stop Recording' button is used the commands are recorded as a macro that performs the same action as recorded. Using

this macro to create a Python script does not require previous knowledge in programming of the user.

3. **Command Line Interface (CLI):** It is easy to access Abaqus CLI which is located in a section of the Abaqus window beneath the Abaqus Viewport. It works in the same way as windows command prompt. It is easy to execute an Abaqus command by entering the command and pushing 'enter'. Basic knowledge of Python syntax is required to use CLI to perform a task. It is, however, easiest way to perform single line command rather than creating and analysing a part or complex code.

6.3.3 Parametric model definition

The first step in modelling a welded joint is to specify the (x, y) coordinates of the main points. With reference to Figure 6-9 the relationship between the parameters k_1 , weld toe radius and weld toe angle in the upper part of the joint is the following:

$$\tan(\alpha_u/2) = \frac{k_1}{\rho_u} \quad \Rightarrow \quad k_1 = \rho_u \tan(\alpha_u/2) \quad (6.1)$$

Similarly for the lower part of the joint

$$\tan(\alpha_l/2) = \frac{k_2}{\rho_l} \quad \Rightarrow \quad k_2 = \rho_l \tan(\alpha_l/2) \quad (6.2)$$

Whilst the relation between the parameter P and the weld toe angle can be expressed as:

$$\tan \alpha_u = \frac{P_1}{0.5 S} \quad \Rightarrow \quad P_1 = 0.5 S \tan(\alpha_u) \quad (6.3)$$

Additionally, for the lower part:

$$\tan \alpha_l = \frac{P_2}{0.5 S} \quad \Rightarrow \quad P_2 = 0.5 S \tan(\alpha_l) \quad (6.4)$$

The relations between all the parameters such as (k, P, weld toe angle α , weld toe radius ρ , plate thickness t, upper reinforcement h_u and lower reinforcement h_l) are used to determine the (x-y) coordinates of the main points which used to create the geometry of the welded joint as defined in Table 6-3. The next step in the process is to complete the welded joint creation and solve in Abaqus. All parameters, material properties, load and boundary condition are included in the Python script. The details of this Python script are discussed in the next section.

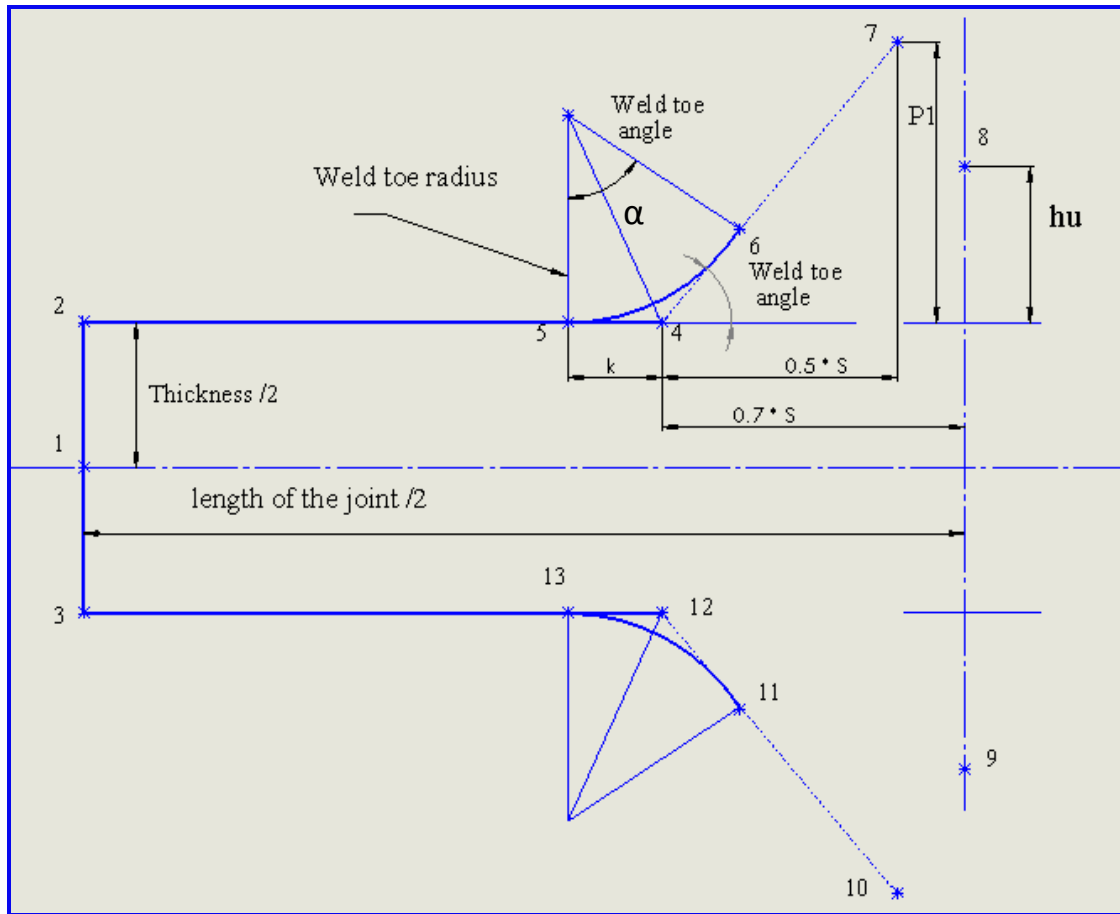


Figure 6-9: Model generation of butt welded joint in Abaqus (Agarwal, 2011)

Table 6-3: Equation for extracting x and y coordinates of welded joint geometry

Points	X- coordinates	Y- coordinates
1	0	0
2	0	$S = t/2$
3	0	$-S = -t/2$
4	$L/2 - 0.7 S$	$S = t/2$
5	$L/2 - 0.7 S - K1$	$S = t/2$
6	$L/2 - 0.7 S + K1 \cos \alpha u$	$S + K1 \sin \alpha u$
7	$L/2 - 0.2 S$	$S + P1$
8	$L/2$	$S + hu$
9	$L/2$	$-S - hl$
10	$L/2 - 0.2 S$	$-S - P2$
11	$L/2 - 0.7 S + K2 \cos \alpha l$	$-S - K2 \sin \alpha l$
12	$L/2 - 0.7 S$	$-S = -t/2$
13	$L/2 - 0.7 S - K2$	$-S = -t/2$

Where t is the thickness of the joints.

Abaqus Script

In Appendix B, a brief description of each part of the Python script is presented. The lines starting with the (#) symbol are treated as comments by the interpreter. The layout of the script is divided into blocks or chunks of code clearly demarcated by:-

```
#####  
# Comment describing the block of code#####
```

The script follows these steps:

1. Initialisation (import required modules).
2. Define the Geometrical Parameters of the Joints
3. Define the x and y Coordinates of Main Points
4. Create the Part
5. Define the Materials
6. Create Section and Section Assignment; Define the Materials
7. Create the Assembly
8. Create Steps
9. Apply Load
10. Apply Boundary Conditions
11. Mesh
12. Create and Run the Job

6.3.4 Parametric model test

In order to investigate the validation of the parametric model the results of two case studies with different parameters are presented below.

Figure 6.10 indicates the stress contours and mesh distribution of a welded joint under tensile loading with the following parameters: weld toe radius = 0.5 mm, weld toe angle = 15°, upper and lower reinforcement = 0.8 mm), whilst Figure 6.11 with the following parameters: weld toe radius = 0.8 mm, weld toe angle = 20°, upper and lower reinforcement = 1.1 mm).

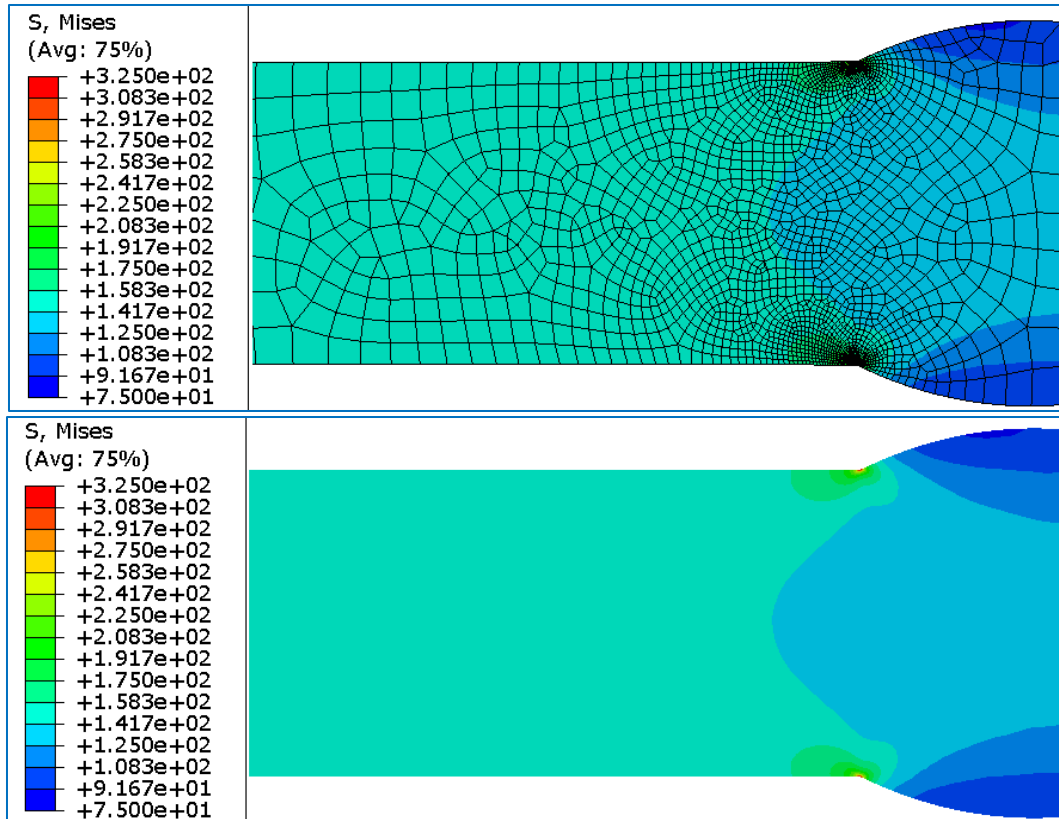


Figure 6-10: Mesh distribution and stress contours of case-1

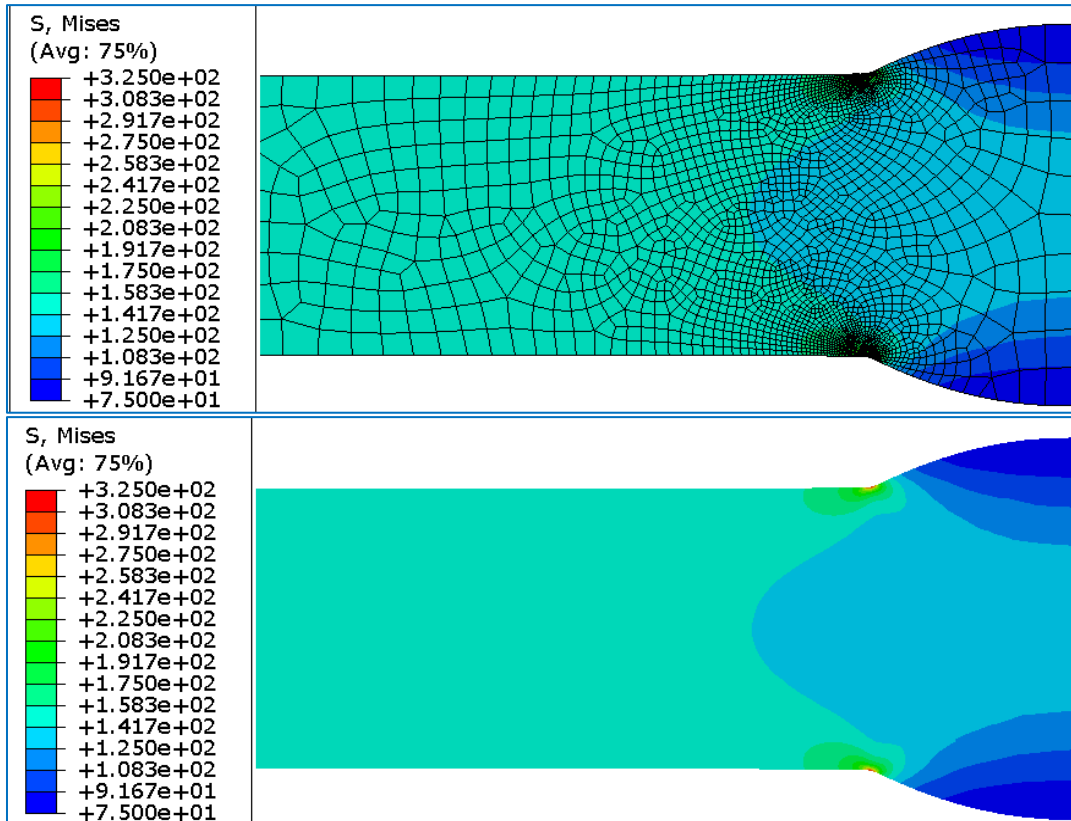


Figure 6-11: Mesh distribution and stress contours of case-2

6.4 Parametric Model Results

In a welded joint, the maximum stress occurs at the weld toe radius and will be larger than the nominal stress at the same cross section area. The stress concentration factor (SCF) can be expressed by the ratio of the maximum stress to the nominal stress; according to Gere (2001), the stress concentration factor is

$$k_t = \frac{\sigma_{max}}{\sigma_{nom}} \quad (6.5)$$

A parametric study was carried out by using the Python script presented in Section 6.3.3 to evaluate the effect of main geometric parameters with two different load cases on SCF. The range of these parameters is presented in Table 6.4.

Table 6-4: Range of geometric parameters

Geometric parameters	Weld toe radius - ρ (mm)	Weld toe angle - α (degree)	Upper reinforcement h_u (mm)	Lower reinforcement h_l (mm)
Tensile load	[0.2-2]	[10-25]	[0.8-1.2]	[0.8-1.4]
Bending load	[0.2-2]	[10-25]	[0.8-1.2]	[0.8-1.4]

The series of finite element simulations can be classified according to load case as

Case –a welded joint under tensile load

The parametric study was performed by evaluating particular parameter variations at a time while the others remained constant. The parameter values of this case are indicated in Table 6.5. From this table, series A represents the study of the effect of weld toe radius on SCF, whilst series B indicates the effect of weld toe angle on SCF. Additionally, series C, D and E show the effect of upper and lower reinforcement with three different values of weld toe angle.

The variation of SCF with weld toe radius is presented in Figure 6.12. It is clear from this figure that increasing weld toe radius will reduce the stress concentration factor, because the higher value of weld toe radius means a smooth profile of the material shape in the intersection point between the base metal and the filler material.

Figure 6.13 indicates the relation between weld toe angle and SCF. It is easy to explain that the higher value of weld toe angle gives high SCF because the increase of weld toe angle (see Figure 6.2) will cause high stress raiser area at the weld toe radius.

The relation between the SCF and upper and lower reinforcement with three different values of weld toe angle (15, 20 and 25) are presented in Figure 6.14. It may be concluded from this figure that the increase of upper and lower reinforcement at the same time and decrease of the weld toe angle will lead to a reduction of the stress concentration factor.

Case –b welded joint under bending load

The geometrical parameters for the second case are tabulated in Table 6.6. From this table the study of the effect of weld toe radius on the stress concentration factor is presented in series A, the relation between weld toe angle and SCF is indicated in series B, whilst series C represents the variation of lower reinforcement with a single value of weld toe angle (20). Finally, series D shows the variation of upper reinforcement with one value of weld toe angle (15).

Figure 6.15 represents the relation between weld toe radius and SCF. The relation between weld toe angle and SCF is indicated in Figure 6.16, the relation between lower and upper reinforcement is indicated in Figures 6.17.

It is clear that there is no significant difference in the behaviour of the welded joint under tensile and bending cases.

Table 6-5: Geometric parameters of welded joint under tensile load

Model		Dimension					Von-Mises stress (Mpa)	K_t	variable
series	No.	t (mm)	hu (mm)	hl (mm)	α (°)	ρ (mm)			
A	1	6	0.8	1.4	15	0.2	410	2.73	Weld toe radius (ρ)
	2	6	0.8	1.4	15	0.5	340.8	2.27	
	3	6	0.8	1.4	15	1	296	1.97	
	4	6	0.8	1.4	15	1.5	274.2	1.83	
	5	6	0.8	1.4	15	2	259	1.73	
B	6	6	1.2	1.2	10	1	192.1	1.28	Weld toe angle (α)
	7	6	1.2	1.2	15	1	267.5	1.78	
	8	6	1.2	1.2	20	1	323.3	2.16	
	9	6	1.2	1.2	25	1	356.4	2.38	
C	10	6	0.8	0.8	15	1	302.2	2.01	Upper and lower reinforcement
	11	6	0.9	0.9	15	1	293.3	1.96	
	12	6	1	1	15	1	284.7	1.90	
	13	6	1.1	1.1	15	1	275.8	1.84	
	14	6	1.2	1.2	15	1	266	1.77	
D	15	6	0.8	0.8	20	1	351	2.34	
	16	6	0.9	0.9	20	1	343.5	2.29	
	17	6	1	1	20	1	337.5	2.25	
	18	6	1.1	1.1	20	1	330.9	2.21	
	19	6	1.2	1.2	20	1	325.4	2.17	
E	20	6	0.8	0.8	25	1	390.2	2.60	
	21	6	0.9	0.9	25	1	379	2.53	
	22	6	1	1	25	1	369	2.46	
	23	6	1.1	1.1	25	1	358.6	2.39	
	24	6	1.2	1.2	25	1	352.5	2.35	

Where:

t	Plate thickness (mm)	α	Weld toe angle (°)
hu	Upper reinforcement (mm)	ρ	Weld toe radius (mm)
hl	Lower reinforcement (mm)	K_t	Stress concentration factor

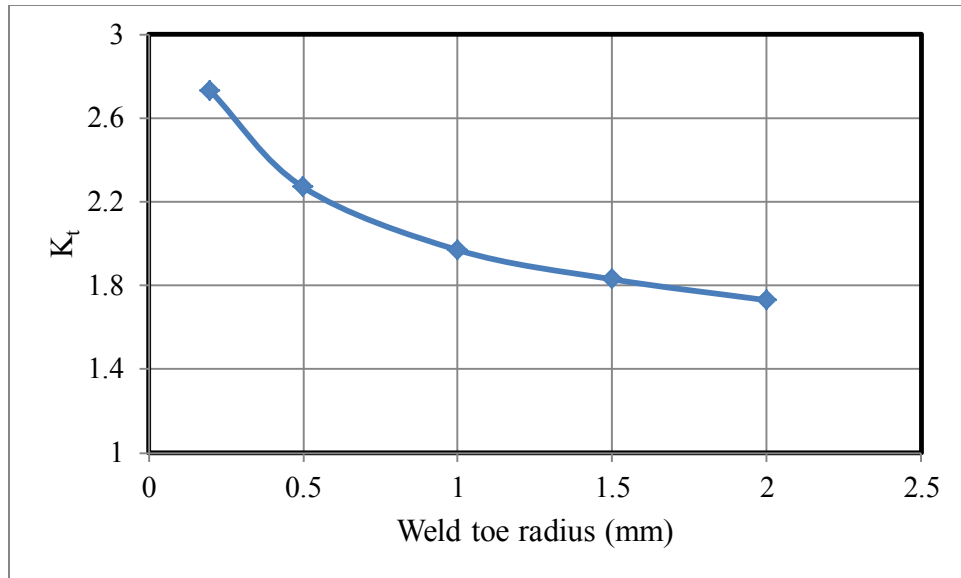


Figure 6-12: Relation between weld toe radius and stress concentration factor (SCF)

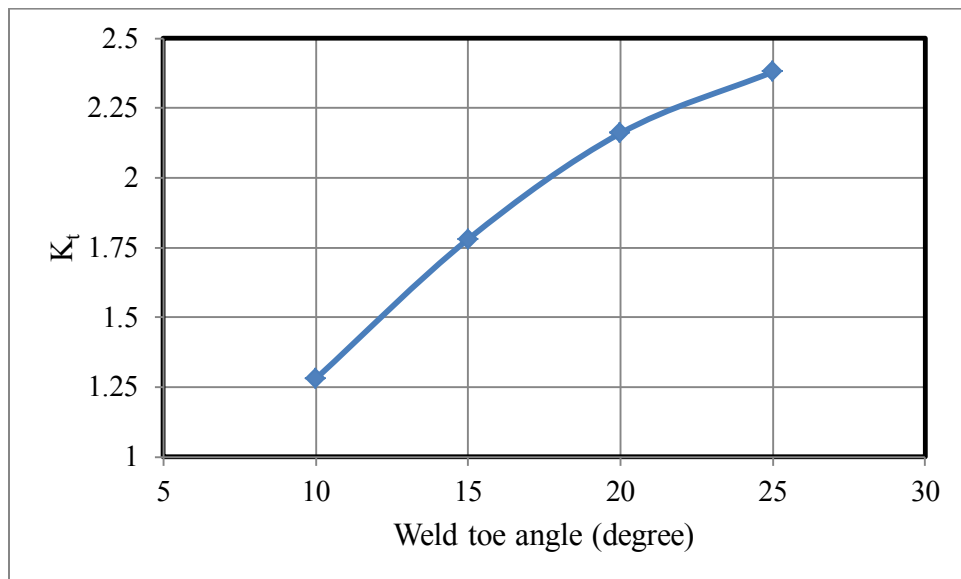


Figure 6-13: Relation between weld toe angle and stress concentration factor (SCF)

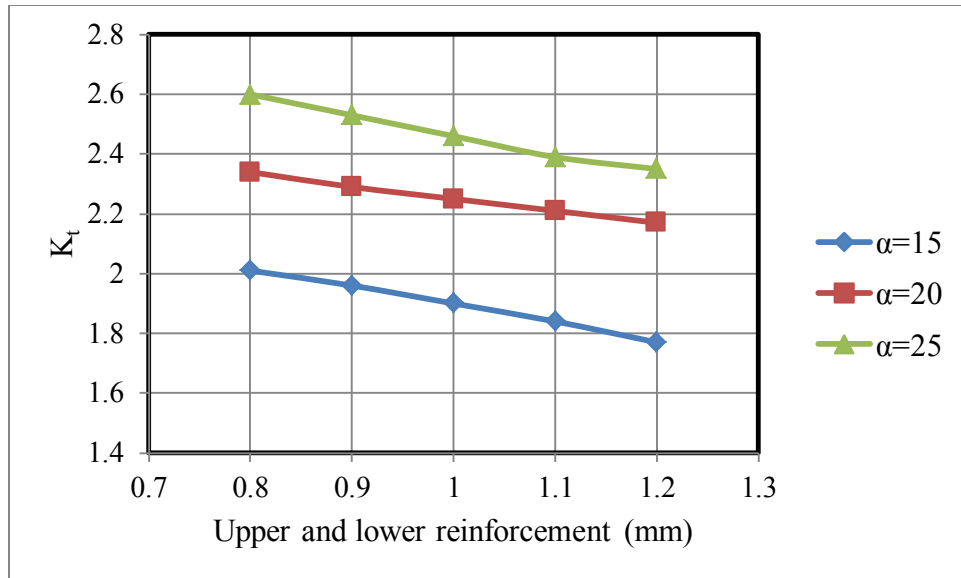


Figure 6-14: Relation between upper, lower reinforcement and stress concentration factor (SCF)

Table 6-6: Geometric parameters of welded joint under bending load

Model		Dimension					Von-Mises stress (Mpa)	K_t	variable
series	No.	t (mm)	h _u (mm)	h _l (mm)	α (°)	ρ (mm)			
A	1	6	0.8	1.4	15	0.2	355.2	2.37	Weld toe radius (ρ)
	2	6	0.8	1.4	15	0.5	302.9	2.02	
	3	6	0.8	1.4	15	1	246	1.64	
	4	6	0.8	1.4	15	1.5	235.5	1.57	
	5	6	0.8	1.4	15	2	229.7	1.53	
B	6	6	1.2	1.2	10	1	176.2	1.17	Weld toe angle (α)
	7	6	1.2	1.2	15	1	233.6	1.56	
	8	6	1.2	1.2	20	1	272.9	1.82	
	9	6	1.2	1.2	25	1	293.5	1.96	
C	10	6	0.8	0.8	15	1	260.2	1.73	Upper and lower reinforcement
	11	6	0.9	0.9	15	1	253.4	1.69	
	12	6	1	1	15	1	246.7	1.64	
	13	6	1.1	1.1	15	1	240.3	1.60	
	14	6	1.2	1.2	15	1	233.2	1.55	
D	15	6	0.8	0.8	20	1	297.6	1.98	
	16	6	0.9	0.9	20	1	286.5	1.91	
	17	6	1	1	20	1	280.8	1.87	
	18	6	1.1	1.1	20	1	276.5	1.84	
	19	6	1.2	1.2	20	1	272.8	1.82	
E	20	6	0.8	0.8	25	1	327.3	2.18	
	21	6	0.9	0.9	25	1	313.5	2.09	
	22	6	1	1	25	1	299.6	2.00	
	23	6	1.1	1.1	25	1	294.4	1.96	
	24	6	1.2	1.2	25	1	291	1.94	

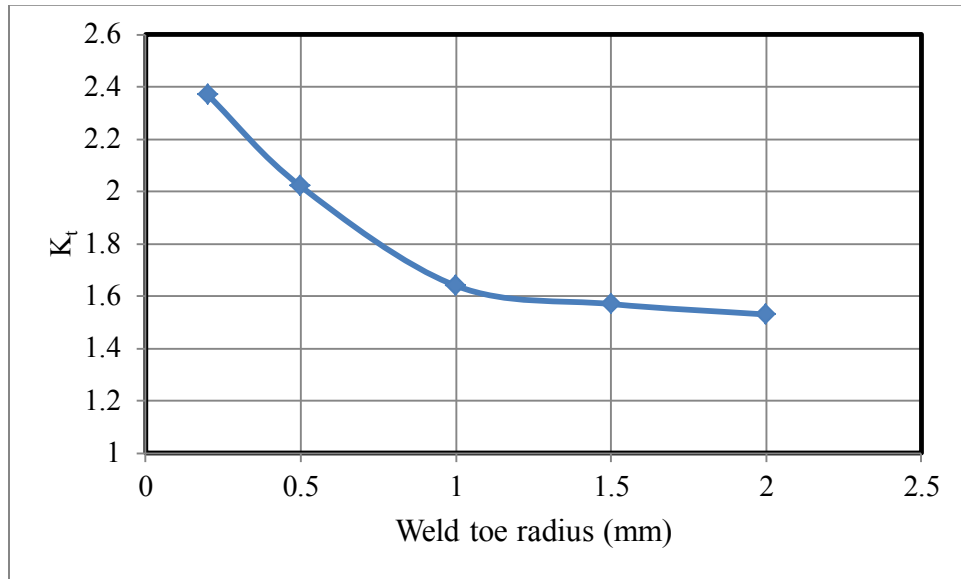


Figure 6-15: Relation between weld toe radius and stress concentration factor (SCF)

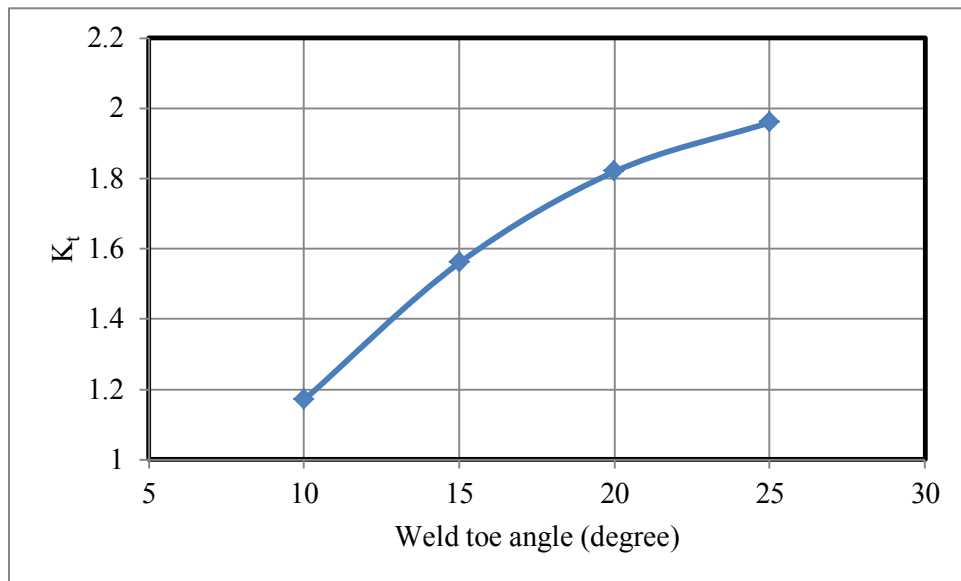


Figure 6-16: Relation between weld toe angle and stress concentration factor (SCF)

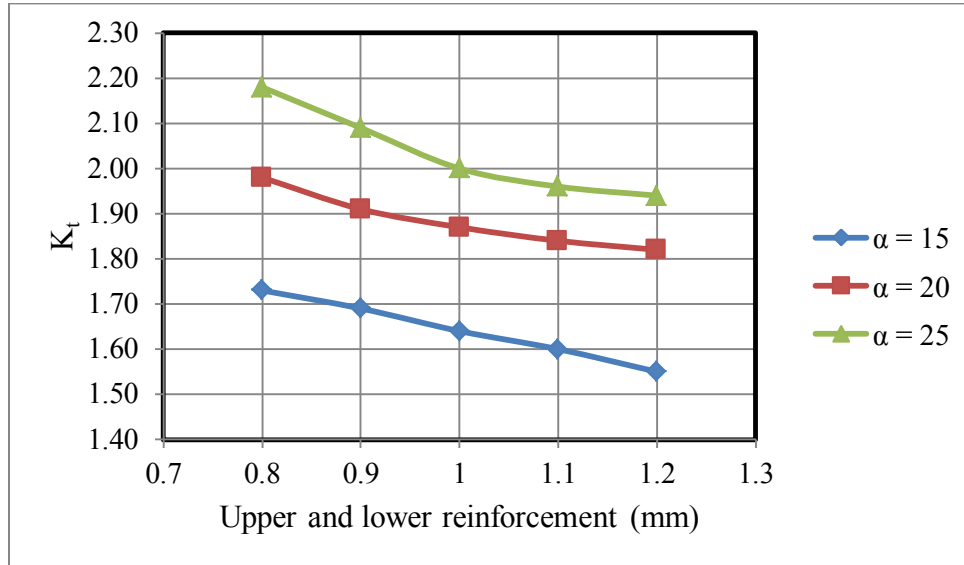


Figure 6-17: Relation between upper, lower reinforcement and stress concentration factor (SCF)

6.5 Conclusion

A parametric model has been generated and has been shown to produce consistent results over a range of geometries.

The relation between geometrical parameters (weld toe radius, weld toe angle, upper and lower weld reinforcement) and stress concentration factor has been investigated and the results indicate the following:

- Increasing the weld toe radius will reduce the stress concentration factor.
- Increasing the weld toe angle will increase the stress concentration factor.
- Regarding the upper and lower reinforcement, increasing the upper and lower reinforcement will reduce the stress concentration factor.

Chapter 7

Experimental Work

Chapter Seven: Experimental Work

This chapter details the experimental work conducted during this study.

Figure 7.1 shows the general steps of the experimental study.

The experimental work constitutes the execution of the following examinations and tests of the material:

- Monotonic tensile testing
- Hardness testing
- Geometrical measurement of the welded joints
- Fatigue testing

The material used during this study was hot rolled structural steel plate (S335JR 27) which is widely used in the petroleum, power generation and chemical industries to fabricate boilers, heat exchangers and tanks. The welding was done by using a laser welding technique. The details of each test are presented in the following sections along with details of the equipment used for the test.

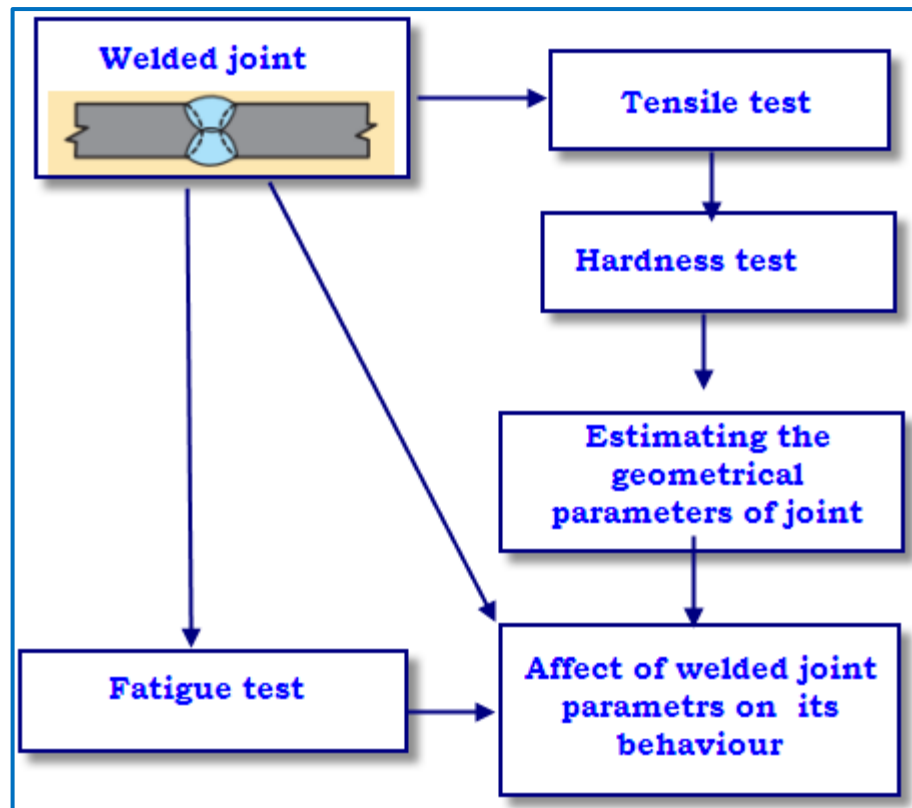


Figure 7-1: Layout of experimental work

7.1 Test Specimen Preparation

After welding, the test specimens were manufactured from the welded plate as shown in Figure 7.2. To avoid heating, and hence interfering with the heat affected zone (HAZ), water jet cutting was used to obtain the test specimens from the welded plate. An additional advantage of this technique was that it generated a clean surface, so the different material regions could be distinguished.

The geometry and dimensions of the tensile test specimens are illustrated in Figure 7.3 and Table 7.1.

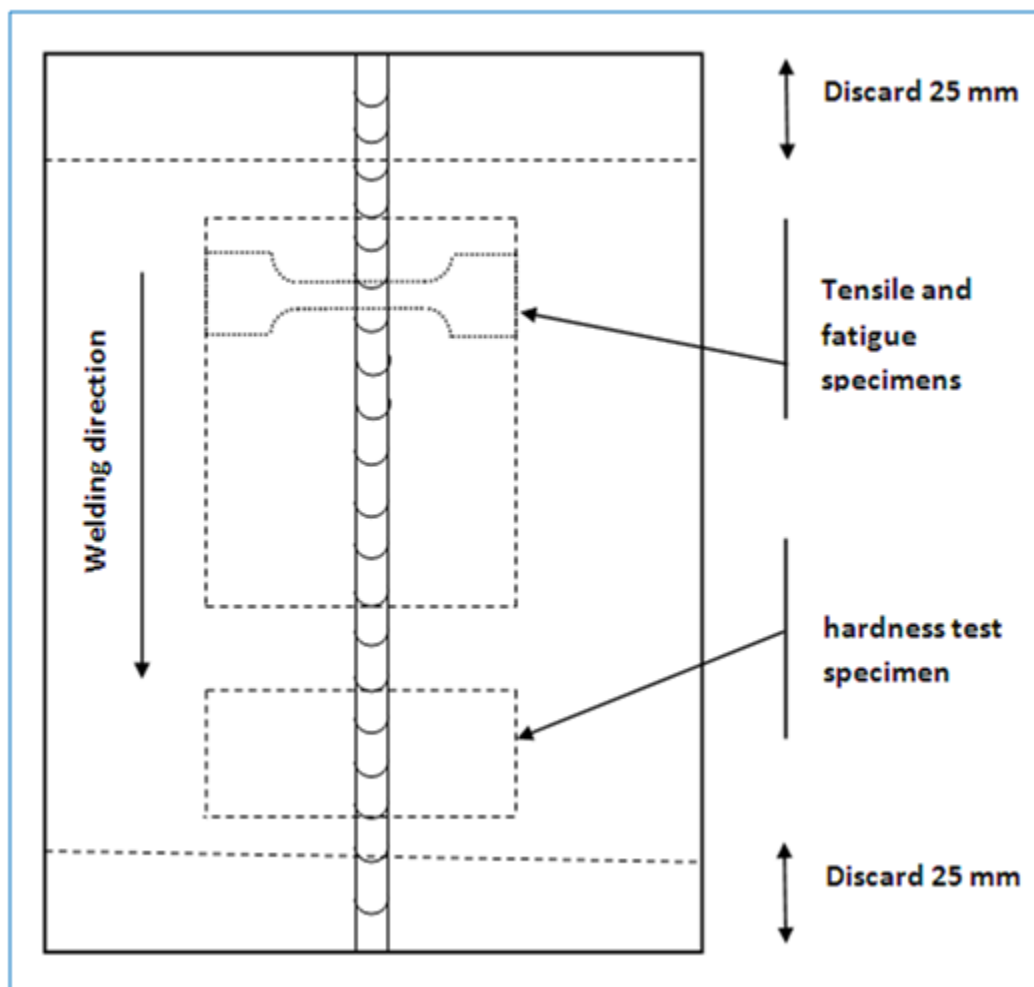


Figure 7-2: Location of test specimens for a butt weld in plate

7.2 Monotonic Tensile Testing

In order to obtain the stress-strain curves and to gather properties of the material, monotonic tensile tests were conducted. All tests were conducted according to EN ISO 6892-1:2009 (British Standard: Metallic materials tensile test). The tests were performed at ambient temperature and in air atmosphere.

The nominal dimensions of the test specimens were 279 mm overall length, 68 mm grip length, 123 mm reduced section, 6.5 mm thickness, as shown in Figure 7.3. From the start until just after yield, an extensometer was used and removed beyond the elastic region to avoid potential failure.

The testing machine was set in displacement control loading mode. The sample was loaded at a speed of 5 mm/min. The dimensions of the 3 tensile test specimens are presented in Table 7.1.

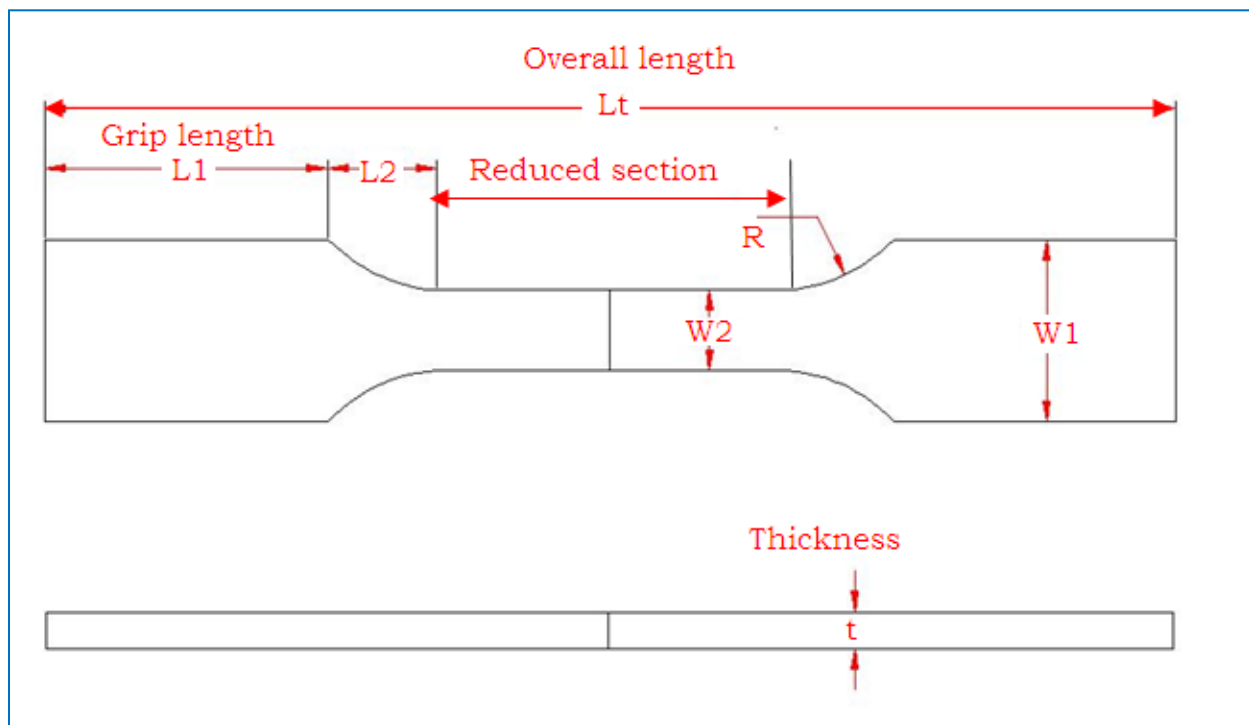


Figure 7-3: Geometry of tensile test specimen (EN ISO 6892-1:2009)

Table 7-1: Dimension of 3 tensile test specimens (EN ISO 6892-1:2009).

Specimen No.	Lt (mm)	L1 (mm)	L2 (mm)	W1 (mm)	W2 (mm)	R (mm)	t (mm)
1	279	69	10	45.1	20.5	22	6.5
2	279	68.5	9.8	44.5	20.3	22.5	6.5
3	280	68	9.5	45	20	21	6.5
average	279.3	68.5	9.76	44.9	20.3	21.8	6.5

The testing machine used was an Instron 3369 table mounted materials testing system with the following characteristics:

- Maximum capacity 50 kN.
- Testing speed range 0.05 to 500 mm/min.
- Integrated digital closed-loop control and data acquisition electronics.

This machine is shown in Figure 7.4.

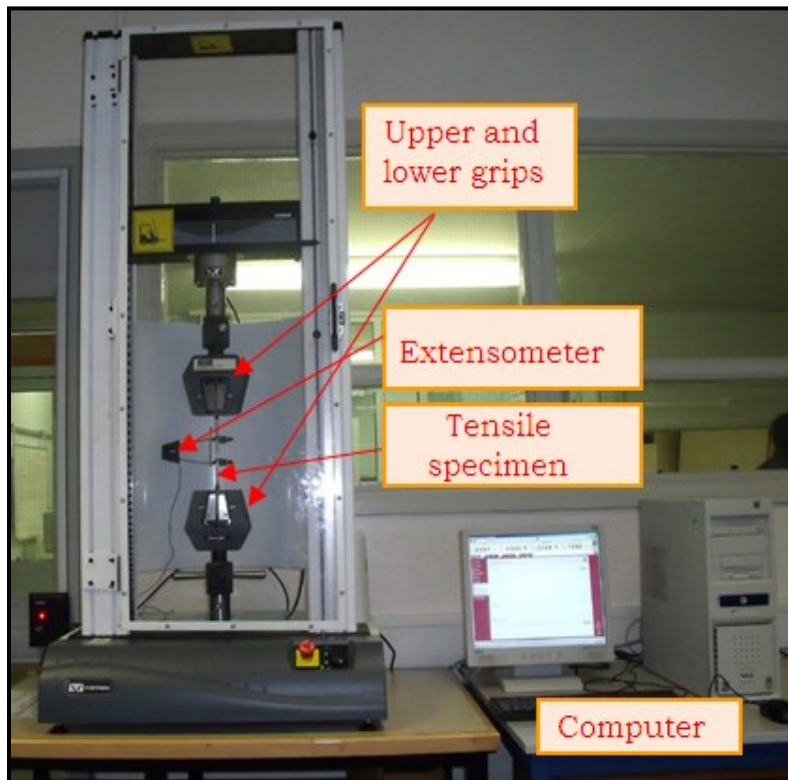


Figure 7-4: Instron (3369) tensile testing machine

The mechanical properties obtained are summarised in Table 7.2: three specimens were used and the result in this table represent the experimental values. The yield strength of samples varied slightly, ranging between 372.5 ± 2.5 MPa, which is less than 0.6% variation in yield strength; there was slight variation in the ultimate tensile strength results within the range 520 ± 10 MPa, which is 1.9% variation.

Table 7-2: Mechanical properties of steel S355JR

Specimen	Yield Strength (MPa)	Ultimate Tensile Strength (MPa)	E (GPa)
1	375	530	207
2	372	520	207
3	370	510	207

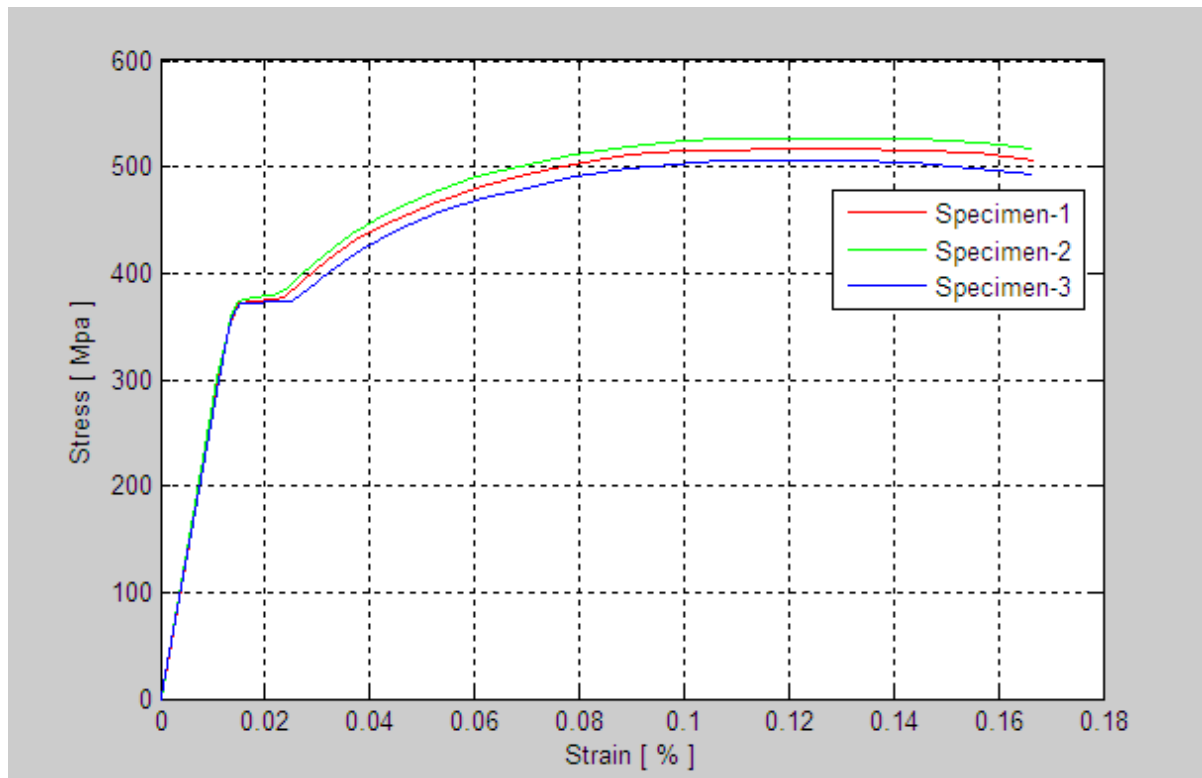


Figure 7-5: Mechanical behaviour of steel S355JR

7.3 Hardness Test

Hardness can be defined as a measure of a material's ability to resist deformation (William, 2003) and can be determined by a macro or micro hardness machine.

A microhardness tester (Buehler, 1600-6100) was used to measure the hardness of the welded joint, as shown in Figure 7.6. Measurements were taken along the longitudinal directions of the joint (1 mm) under the surface in accordance with (EN 288-3); see Figure 7.7. Fine polished samples were tested at load 200 gf for 15 seconds.

Table 7.3: shows the results of the hardness measurements in the different zones; base metal, HAZ and filler material. The mean hardness values were 250 in the weld metal and 264 in the HAZ region. The base metal had a mean hardness value of 225.

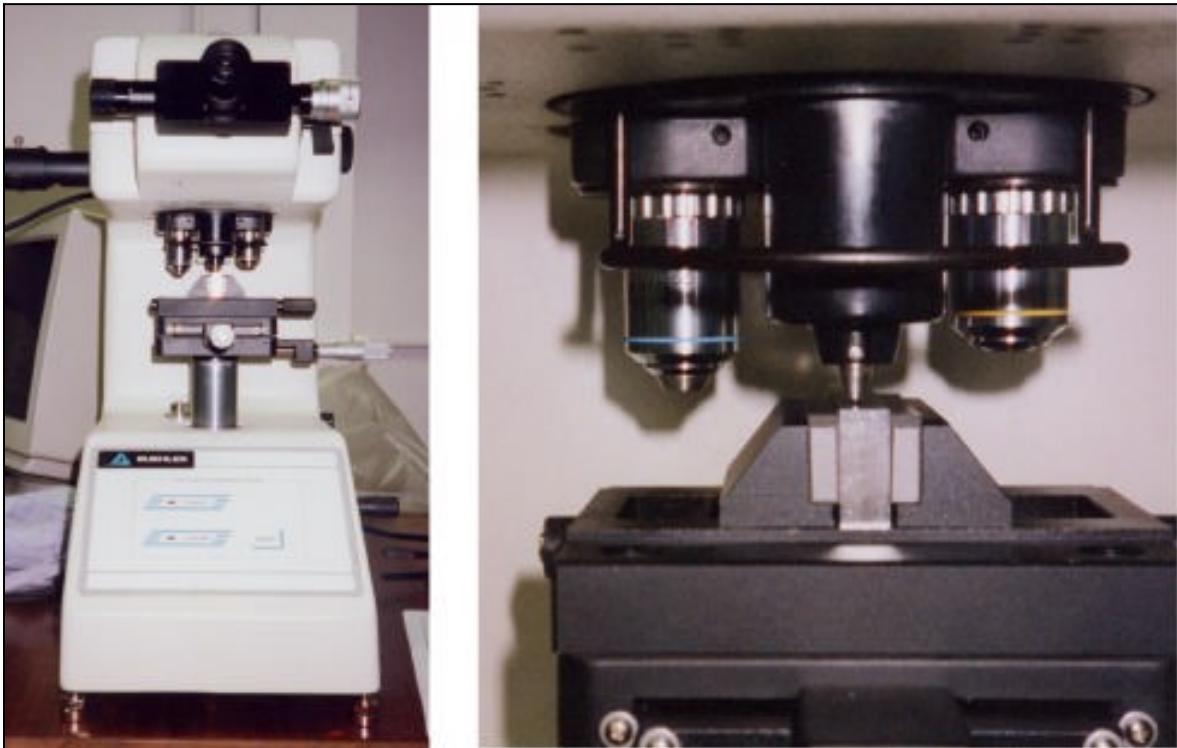


Figure 7-6: Microhardness machine Buehler 1600-6100 (left), and close up of specimen (right).

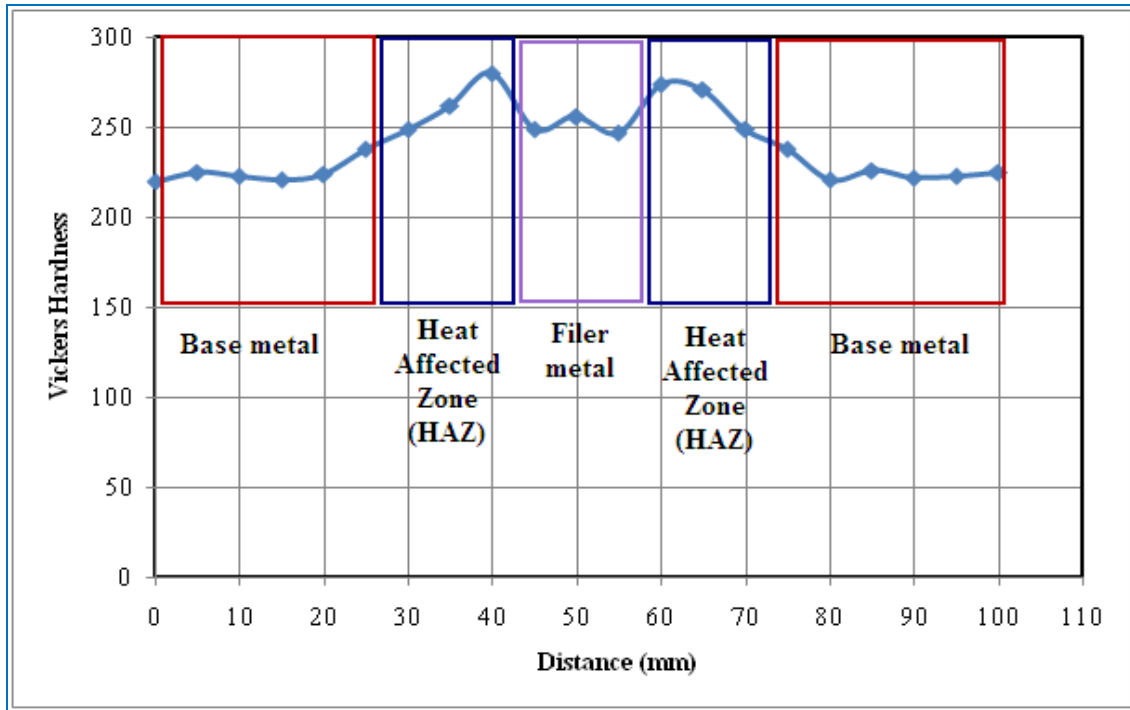


Figure 7-7: Hardness profile of butt welded joint

Table 7-3: Micro hardness results of butt welded joint

Position	Hardness (Vickers)	Maximum value	Mean value
Base metal	220	238	225.2
	225		
	223		
	221		
	224		
	238		
Heat Affected Zone (HAZ)	249	280	263.7
	262		
	280		
Filler metal	249	256	250.7
	256		
	247		
Heat Affected Zone (HAZ)	274	274	264.7
	271		
	249		
Base metal	238	238	225.9
	221		
	226		
	222		
	223		
	225		

7.4 Estimating the Geometrical Parameters of Welded Joints

The geometrical dimensions of a butt welded joint can be divided into four different groups: weld toe angle α , the height of the weld h , the width of weld w , and weld toe radius ρ , as indicated in Figure 7.8. The purpose of this stage of the project was to obtain an estimate of the range of weld dimensions for a practical weld. Accurate measurements of a particular weld were not required.

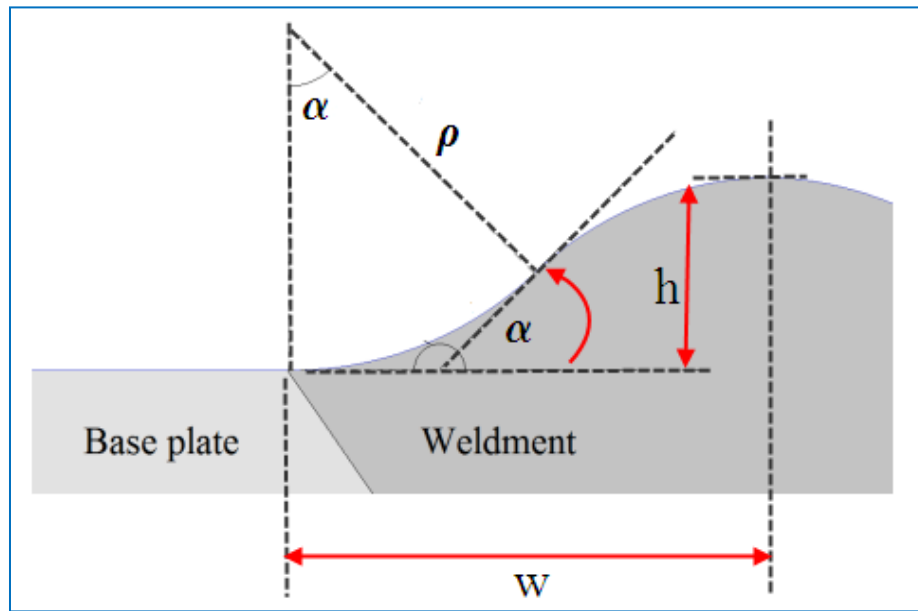


Figure 7-8: Dimension of butt welded joints: weld height h , weld width w , weld toe radius ρ , and weld toe angle α .

7.4.1 Estimation of welded Joint geometrical parameters by using dental moulding method

Dental moulding was used to estimate the welded joint geometry (see Figure 7.9) by taking a negative impression of the weld. This method applied dental cement to the surface of the weld. When the cement had hardened, the mould was removed from the weld and cut into slices. The impression of the weld profile was then photographed and imported into the AutoCAD software as a digital image for further analysis. The dental cement procedure is described below.

Dental moulding procedure

To ensure that the silicone was placed in the best position to handle the casting, a dispensing pistol was used. The equipment used is shown in Figure 7.9.

1- Desired specimen to make casting

- 2- Casting pistol
- 3- Affinis heavy body surface activated cement
- 4- Mixing tips



Figure 7-9: Equipment used in geometrical estimation of welded joints

Step-1: The specimen was arranged with supported strips on each side of the mould region, as presented in Figure 7.10. This prevented migration of silicone away from the weld.

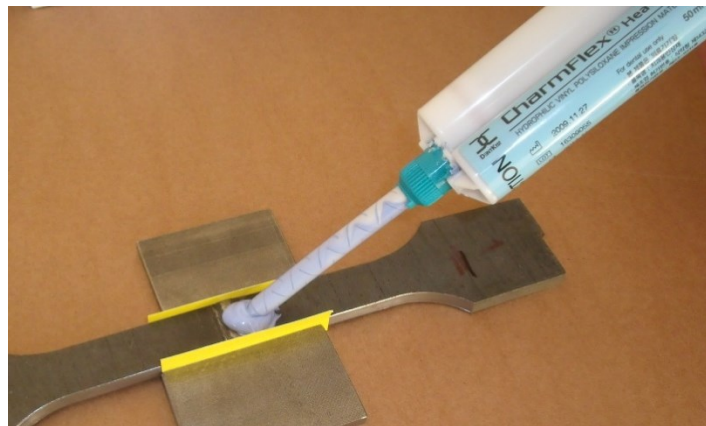


Figure 7-10: Dental moulding cement procedure

Step-2: The silicone was applied using the pistol according to the manufacturer's data sheet (see Figure 7.10).

Step-3: At least 10 minutes were allowed for the silicone to cure.

Step-4: After 10 minutes, the casting process was complete.

Step-5: The dental cement moulding of the weld was marked and sliced at different locations, and then the slices were analysed using the AutoCAD software.

Estimation Results

The estimation results, which indicate the values of weld toe angle α , weld toe radius ρ , upper weld width w_u , lower weld width w_l , upper reinforcement h_u and lower reinforcement h_l , are indicated in Table 7.4. This table also indicates the standard deviation of the estimation.

Figure 7.11 indicates the geometrical parameter estimation by using the dental moulding procedure. It should be noted in this diagram that the weld toe angle is defined using the tangent to the weld face 1 mm above the base metal line. This follows the definition given by (Sechadri, 2006).

As mention before, the accurate measurement of weld geometry are not required, Table 7.4 gives a general idea about the variation of geometrical parameters. From the results in Table 7.4 it is possible to specify the upper and lower limits and standard deviation for geometrical parameters of welded joint which will used in robust optimisation study.

For example the values of weld toe radii vary (0.5-0.61), so the ranges of this parameter in robust optimisation study (0.5-5).

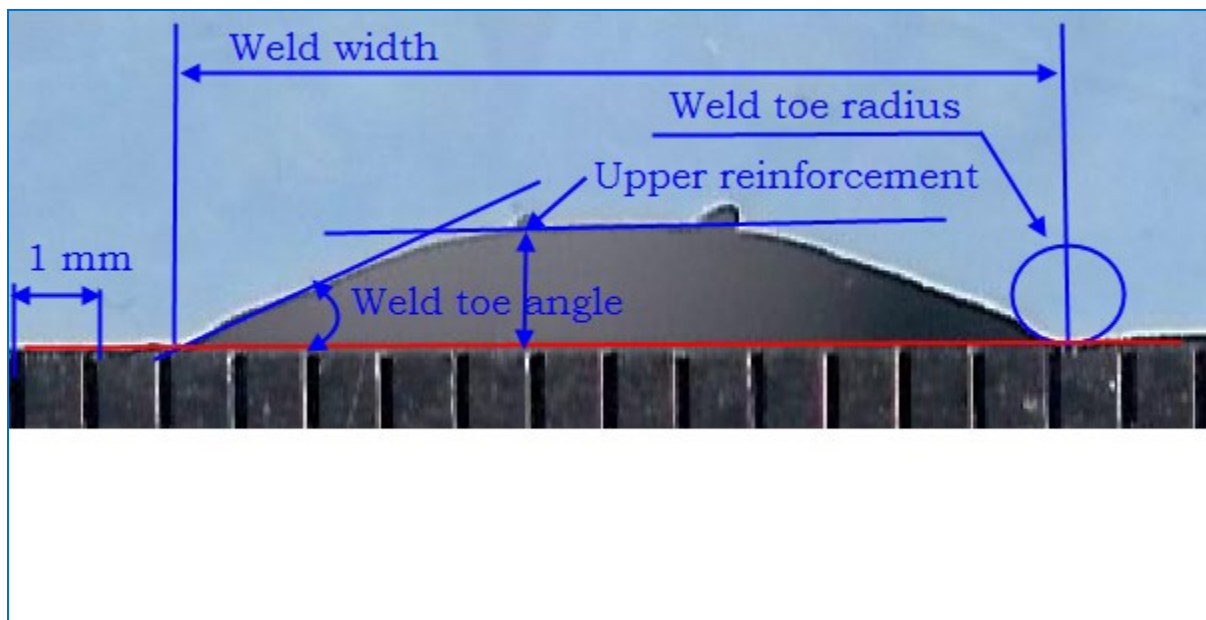


Figure 7-11: Weld joint parameter by using dental moulding

Table 7-4: Estimation results of 11 welded joints

	spe · no.	Weld toe radius				Weld toe angle				Upper width	Upper reinforce.	Lower width	Lower reinforce.
		ρ_1	ρ_2	ρ_3	ρ_4	α_1	α_2	α_3	α_4	w_u	h_u	w_l	h_l
	1	0.629	0.535	0.54	0.64	24	22	23	22	13.5	1.93	9.03	1.11
	2	0.685	0.729	0.74	0.73	25	21	22	21	11.6	2.09	12.15	1.08
	3	0.41	0.5	0.82	0.64	28	27	21	19	11.63	1.68	10.74	1.2
	4	0.504	0.45	0.67	0.62	28	23	19	22	11.83	1.98	10.05	1.005
	5	0.45	0.35	0.58	0.75	25	25	21	18	12.11	1.58	9.88	1.17
	6	0.55	0.66	0.62	0.64	27	25	19	20	12.9	2.29	11.4	1.1
	7	0.59	0.51	0.54	0.55	24	23	20	21	10.35	1.52	11	1.2
	8	0.65	0.63	0.62	0.64	23	22	21	20	11.05	1.78	10.8	1.1
	9	0.42	0.52	0.68	0.76	23	23	22	22	11.3	1.5	11.05	1.2
	10	0.51	0.61	0.68	0.74	25	23	23	23	11.4	1.6	12.2	1.75
11	0.6	0.65	0.61	0.67	27	25	22	21	11.6	1.7	11.36	1.02	
std. Dev.													
		0.094	0.108	0.084	0.065	1.858	1.753	1.401	1.470	0.857	0.25555	0.950	0.20247
mean values													
		0.545	0.558	0.645	0.670	25.36	23.54	21.18	20.81	11.75	1.78636	10.87	1.17591

7.5 Fatigue Test

7.5.1 Test specimen

The dimension of the fatigue test is presented in Figure 7.12. All the specimens were kept in as-weld condition, and the loading direction was transverse to the weld direction.

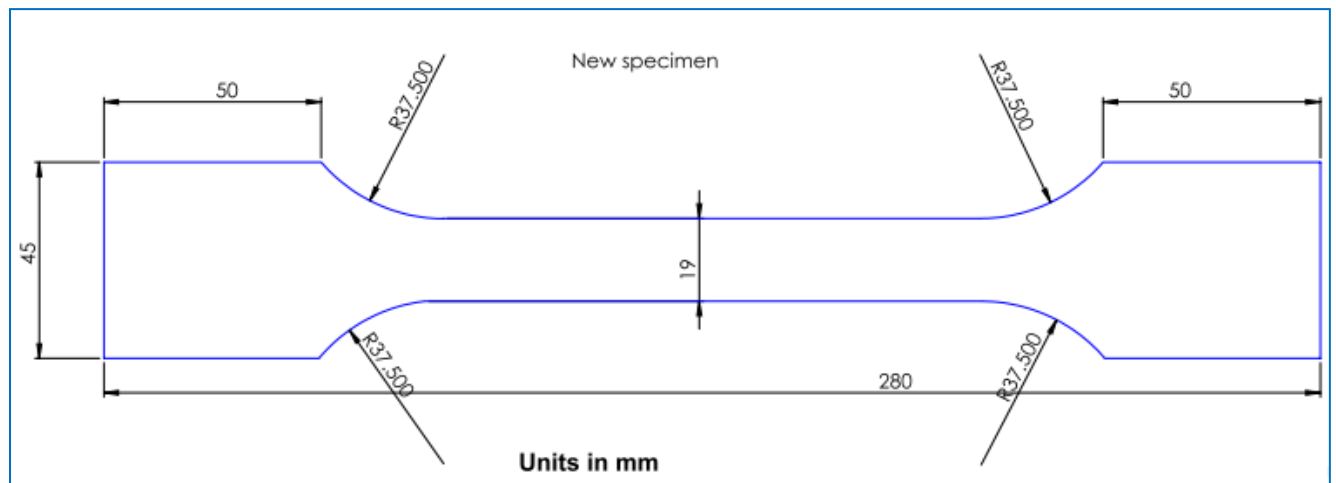


Figure 7-12: Fatigue test specimen

7.5.2 Fatigue testing

Fatigue testing was performed at room temperature using an Instron hydraulic fatigue testing machine with corresponding software (see Figure 7.13).

The specimens were subjected to tensile load cycled sinusoidally at a frequency of 1 Hz. The stress ratio was held constant at ($R=0.1$) for all specimens tested. All the tests were conducted until failure. The results of the fatigue tests on the welded joints are indicated in Table 7.5.

The S-N curve of stress range ($\Delta\sigma$) versus the corresponding number of cycle to failure (N) is plotted in Figure 7.14 to indicate the fatigue properties of welded joints. In this figure the obtained results are compared with results extracted from Pigneaux (2002) for butt welded joints in tension at a stress ratio ($R=0.1$), but with a different type of welding.

A sample of the fatigue load calculation for the fatigue experiment is presented in Appendix-C.

Figure 7.15 shows the specimens after failure. It is seen that the fatigue cracks initiated at the transition between the plate and the weld. Figure 7.16 shows the fatigue crack front. It is clear to see in this figure that fatigue crack started from the weld toe and grew through thickness prior to final failure.

Referring back to (Figure 2.3), it is easy to distinguish the three main stages of crack growth:

- Early stage of propagation: the crack propagates along the weld toe.
- In the second stage the crack propagates perpendicular to the face of the plate.
- The final failure involves rapid crack propagation at about 45° to the tensile axis.



Figure 7-13: Fatigue testing machine

Table 7-5: Fatigue test results

Specimen No.	Cycles to failure, N	Stress range, $\Delta\sigma$ [MPa]	Mean stress, σ_{mean} [MPa]	Stress amplitude, σ_{amp} [MPa]
1	5.60450e+5	200	122.1	100
2	357430	225	137.47	112.5
3	310911	250	152.75	125
4	263000	275	168.02	137.5
5	103685	310	189.41	155
6	91530	325	198.58	162.5

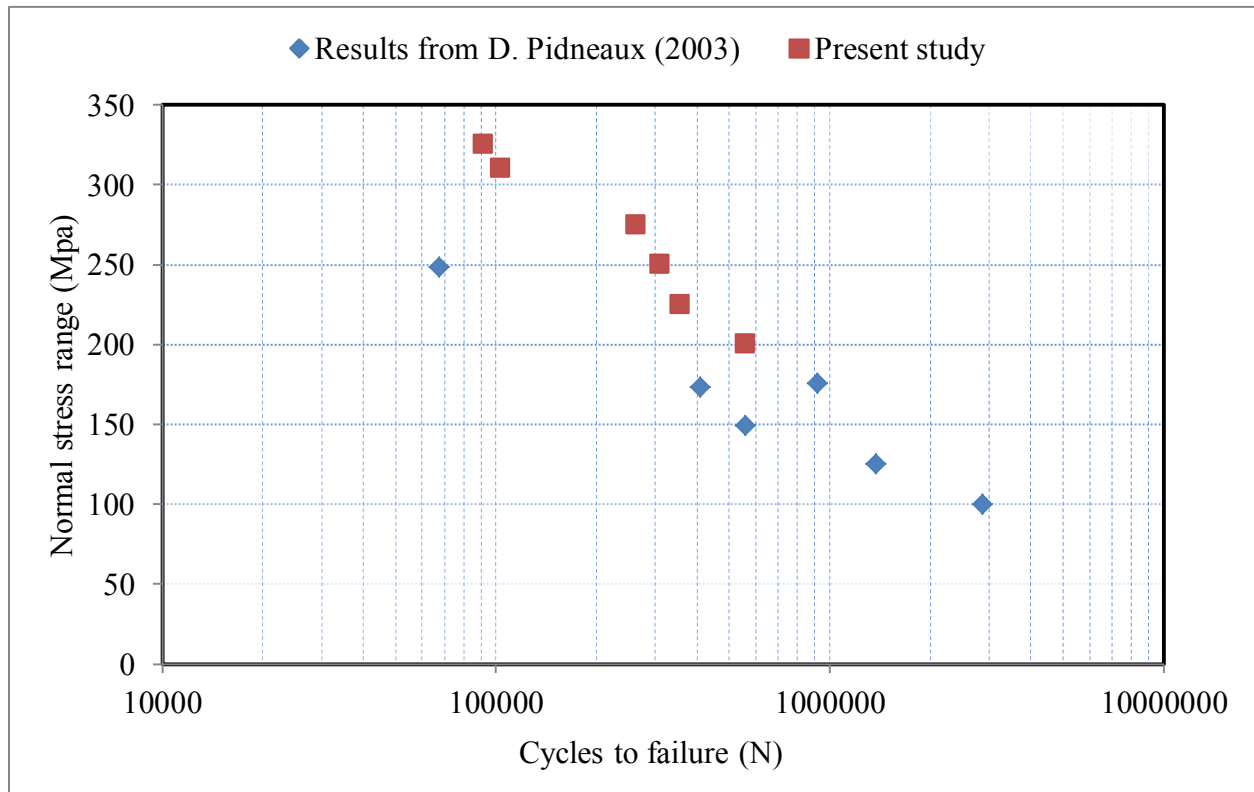


Figure 7-14: Characteristic fatigue strength S-N curves.

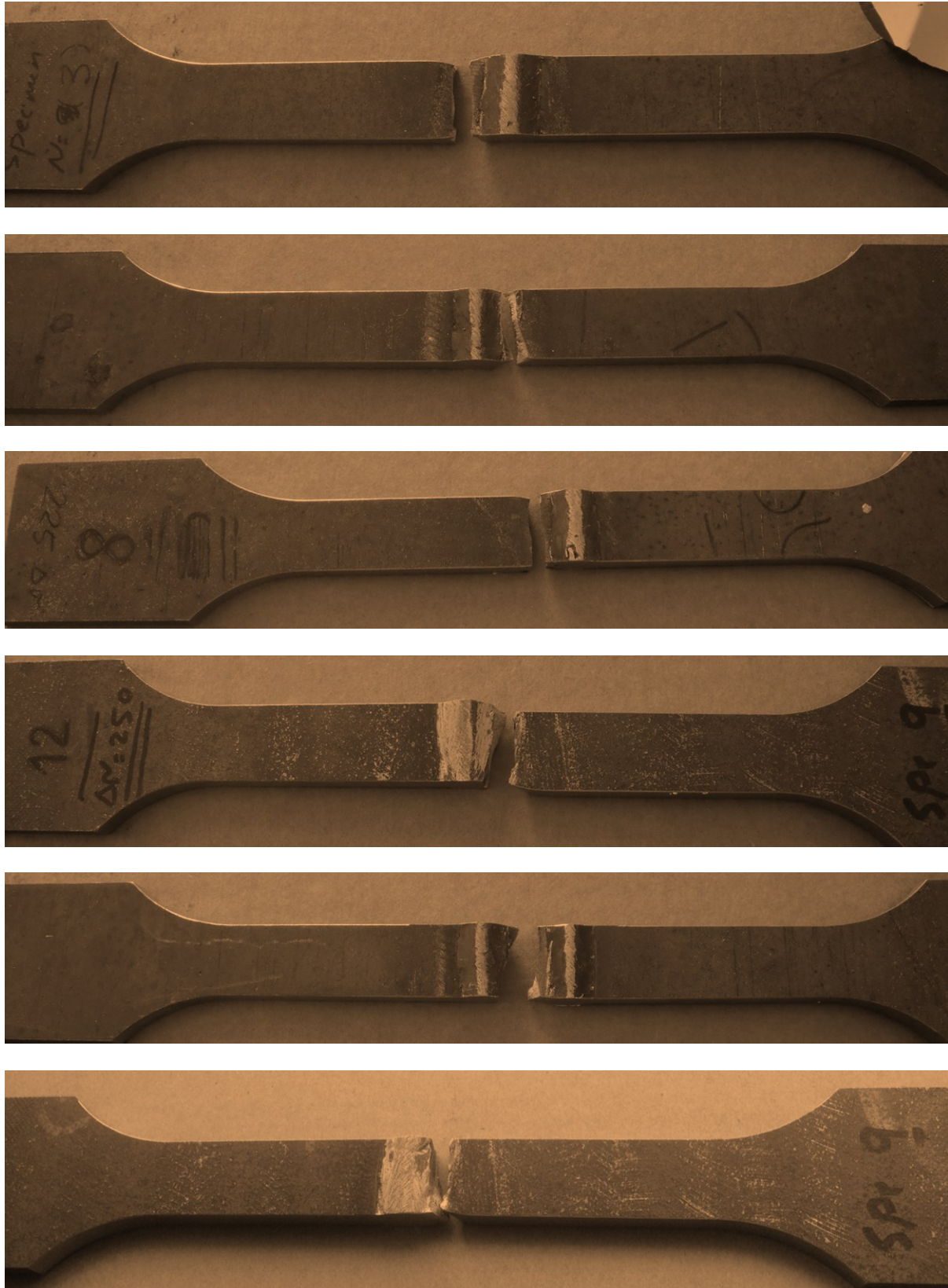


Figure 7-15: Examples of failure specimens

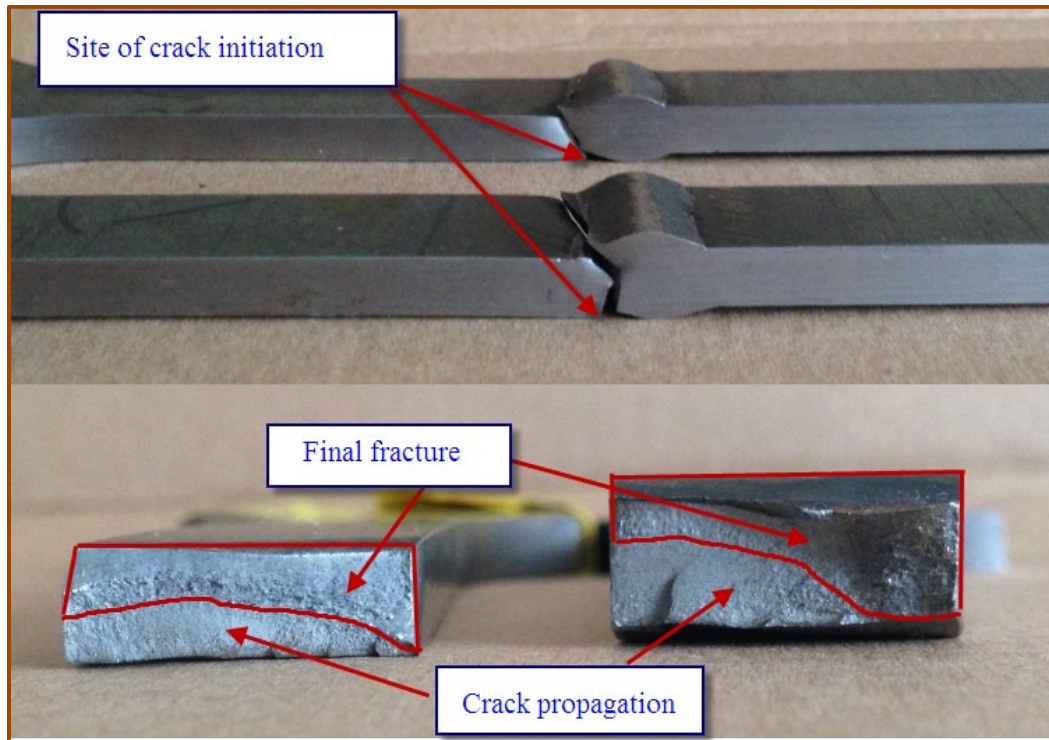


Figure 7-16: Examples of fatigue failure stages

7.6 Conclusion

Different test have been performed during this chapter for the (S335JR 27) hot rolled structural steel.

Three samples subjected to static tensile test the results indicate the variation of results is ranging between 0.6% for yield strength and 1.9% for ultimate tensile strength.

From the hardness test it is clear that the filler material is harder than the base whilst the heat affected zone is the hardest region.

In order to estimate the welded joint geometry, a replica was made using dental mould material. This method is time consuming but gives acceptable results.

Fatigue tests with six specimens under tensile loading showed results that were consistent with published data.

In all the fatigue test specimens, fatigue was seen to initiate from the weld toe. This underlines the importance of the geometry in this area.

Chapter 8



Robust Optimisation of welded joints

Chapter eight: Robust Optimization of Welded Joints

8.1 Introduction

In this chapter traditional and robust optimization of welded joints is introduced, it consists of three main parts. In the first part the integration between Abaqus and modeFRONTIER software has been introduced with two optimization problems. In the first problem a step by step detailed example of multi-objective optimization with two objective functions has been introduced. In this example the geometry and mesh remained constant, only the element properties changed. This allowed the link between modeFRONTIER and Abaqus to be investigated without the complication of geometry and mesh redefinition. The second problem, deals with multi-objective optimization of plate with hole. In this problem the additional, significant complexity of ensuring that a mesh and hence analysis of an acceptable quality has been generated over a range of geometries was investigated.

The second part of this chapter includes the traditional optimization of welded joints. The values of input variables were taken from real values estimated in chapter 7. The result of this part which includes the Pareto solution is analyzed and used in the final part of this chapter.

In the third part the robust optimization of welded joints is performed. Three different cases are studied according to the values of standard deviation of input design variables.

8.2 Integrating between modeFRONTIER and ABAQUS

This section describes a step by step guide for configuring the interface between modeFRONTIER and Abaqus. Two different examples with different parameters have been investigated as follows

1. Multi-objective Optimization of Two-bar Truss Problem (with 2 Objectives function)
2. Multi-objective Optimization of Plate with Hole (with 2 Objectives function)

8.2.1 Multi-objective Optimization of Two-bar Truss Problem

A two beam structure with a rectangular cross-section is loaded with a force F , the structure has an encastre constraint at points A and B (no displacement or rotation), see Figure 8.1.

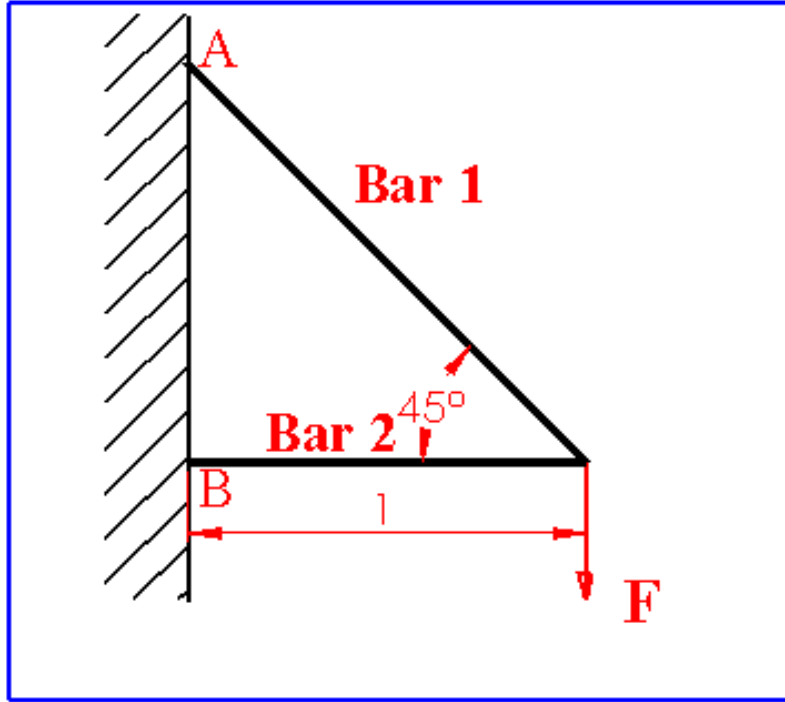


Figure 8-1: A two beam structure with a load F.

The problem has two conflicting objectives; the goal of optimization is to reduce the weight by minimizing the volume V and stress σ under applied load F .

$$\text{Objective 1 } \mathbf{Min} \ V \quad (V = \text{volume}) \quad (8.1)$$

$$\text{Objective 2 } \mathbf{Min} \ \sigma \quad (\sigma = \text{stress}) \quad (8.2)$$

The input variables bound are:

$$20 \leq \mathbf{a_1} \leq 100 \quad [\text{mm}]$$

$$20 \leq \mathbf{b_1} \leq 100 \quad [\text{mm}]$$

$$20 \leq \mathbf{a_2} \leq 100 \quad [\text{mm}]$$

$$20 \leq \mathbf{b_2} \leq 100 \quad [\text{mm}]$$

Where ($\mathbf{a_1}$, $\mathbf{a_2}$) are the width of beam 1 and beam 2, ($\mathbf{b_1}$, $\mathbf{b_2}$) are the height of the cross sections of beam1 and beam2 respectively, with the load F equal top 100 kN.

The final work flow is presented in Figure 8.2. The work flow can be approached in two ways: the data flow, from top to bottom, and process flow from left to right.

The FMOGA-II optimization algorithm was used to solve this example. With 10 iterations, implementing 20 Sobol as the DoE method, the total number of designs tested was 200.

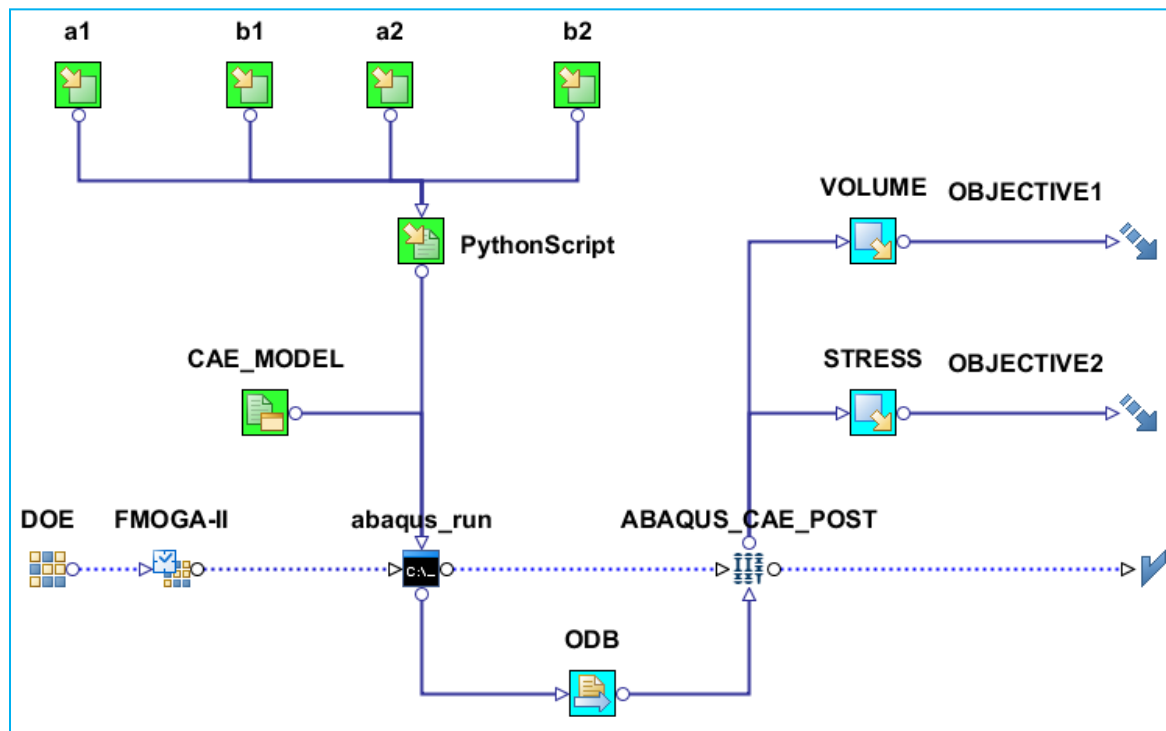



Figure 8-2: Pareto front of the two-bar truss problem.

The Input variable-node  is used to enter the value / interval of the design parameters, the design parameters are set as a constant or variable, Figure 8.3.

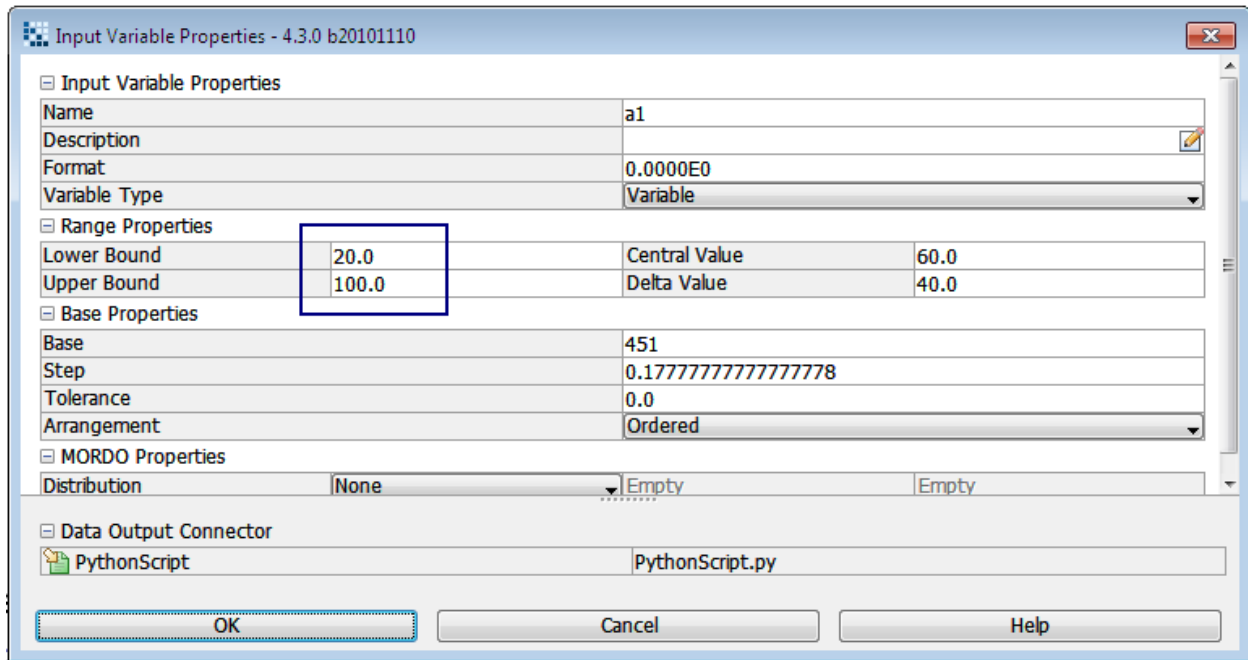



Figure 8-3: Input variable-node Settings, the variable can be set as constant or variable.

The input variables are linked to a Python script (Input file node ) which enables the changing of these variables in Abaqus / CAE model (see Section 6.5.1.). To make modeFRONTIER distinguish what value to change the input variables are tagged in this script, at the button Edit Input File to select the Python script and enter the variables in the flow, see Figure 8.4 and Figure 8.5.

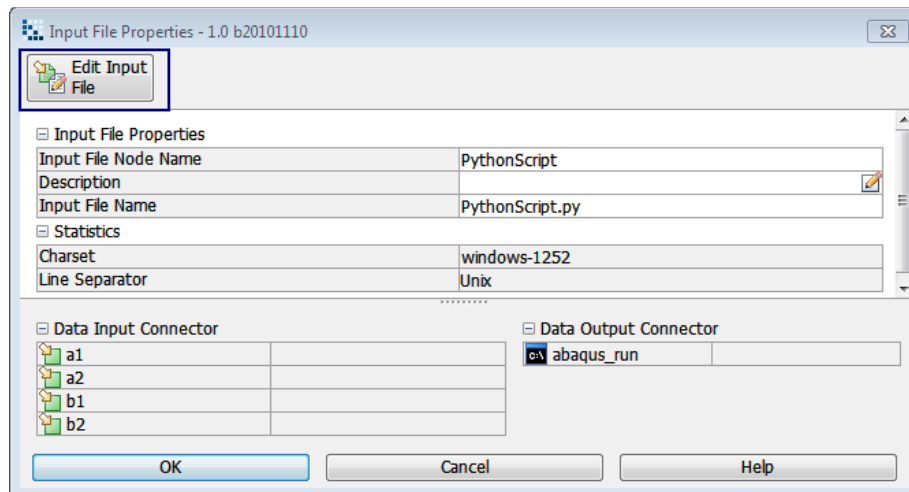


Figure 8-4: Settings for the Input file-node.

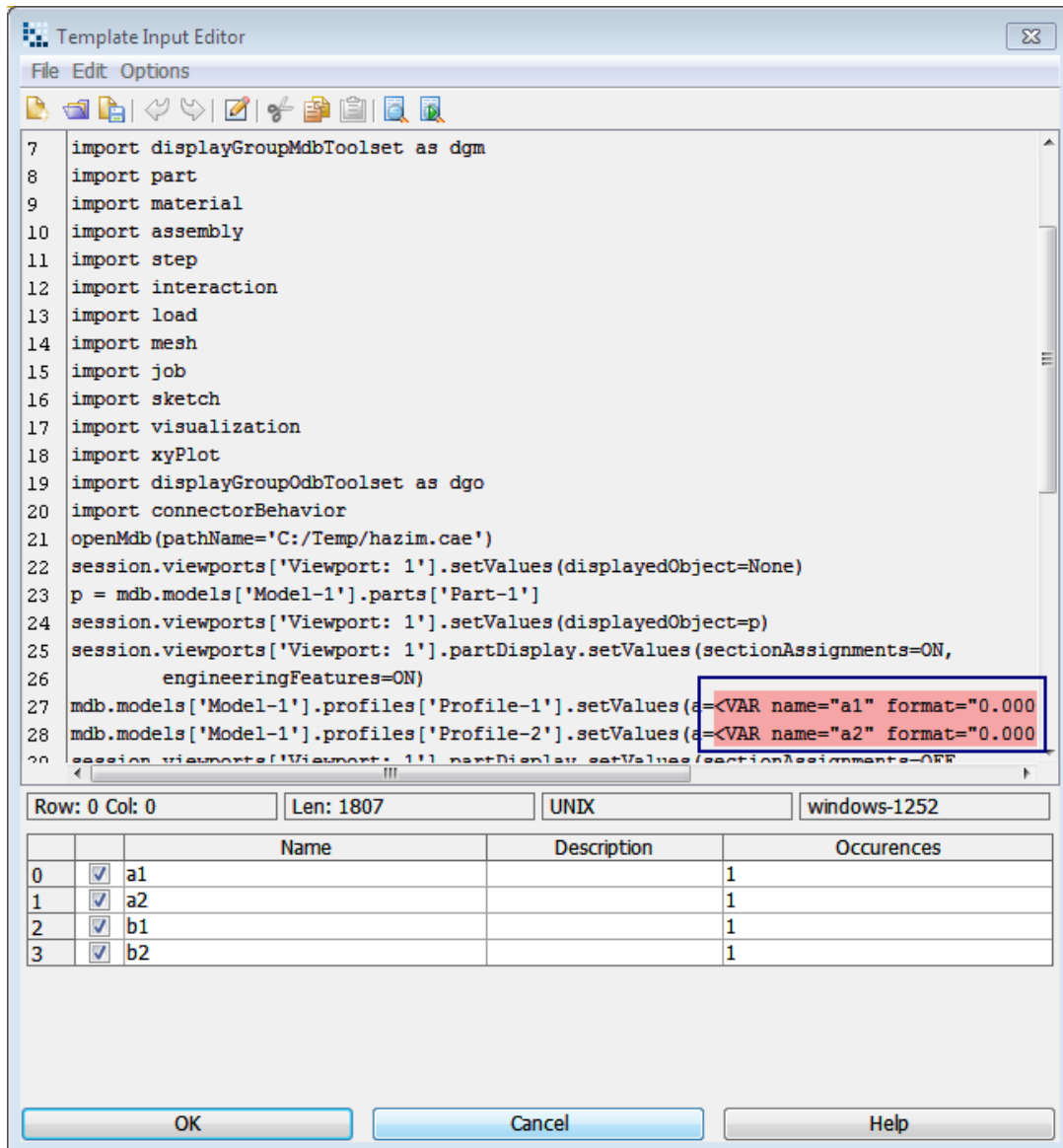



Figure 8-5: Input file-node editor.

To instruct modeFRONTIER to run Abaqus a DOS Batch script node  is generated as shown in Figure 8.6. This node runs the Python script through Abaqus / CAE and this runs Abaqus to generate an output data base, Job-1.odb. This contains the results for the specific design.

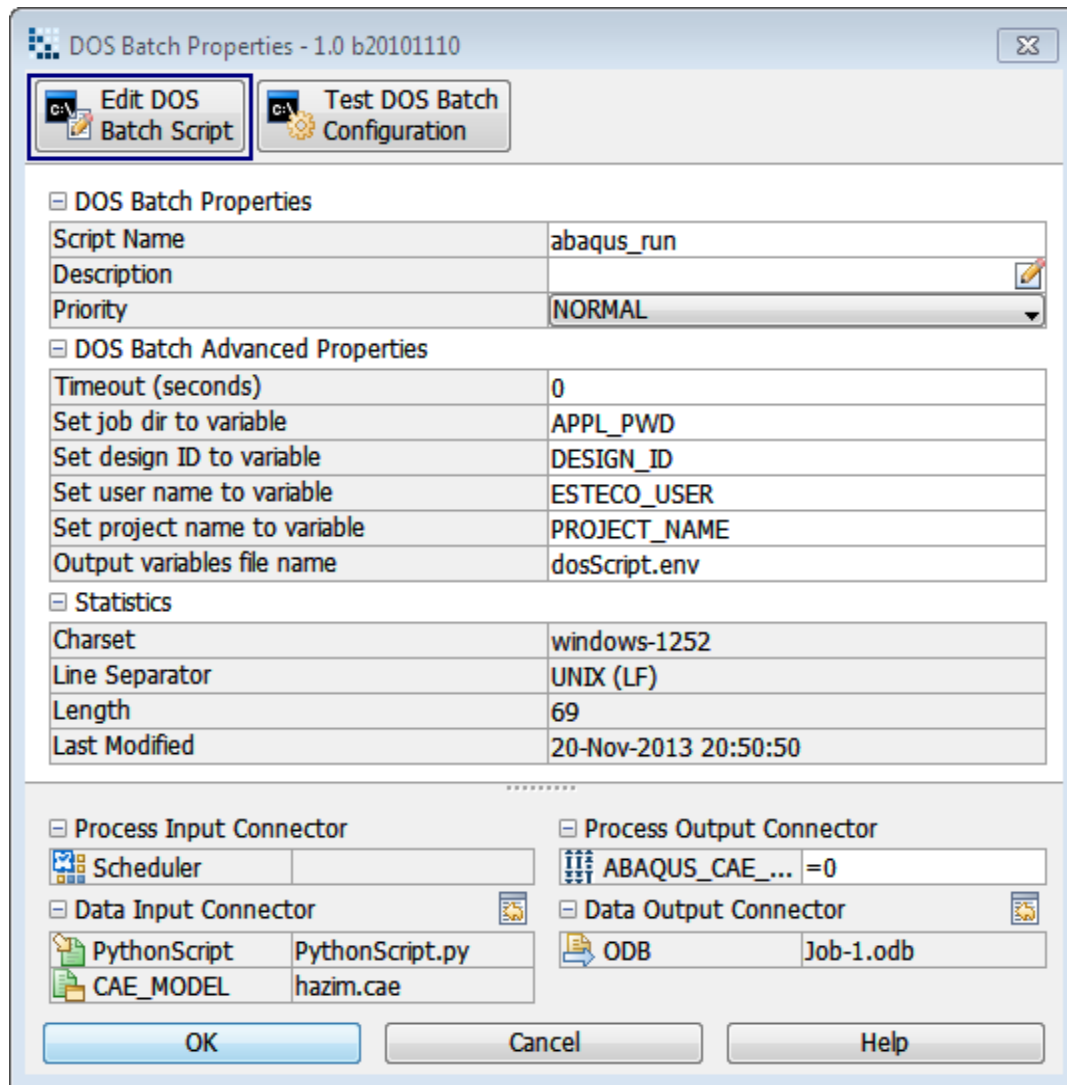
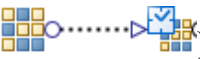



Figure 8-6: DOS-node properties, at the button Edit DOS Batch Script commands are written.

From the scheduler-nodes () where the first node is a DoE -node, the output is fed to the DOS-node. In this example a Sobol sequence was used to generate 10 experimental designs when the FMOGA-II algorithm is used.

In order to extract the required criterion data from the Abaqus output database (odb file) an Abaqus / CAE-POST  node is define.

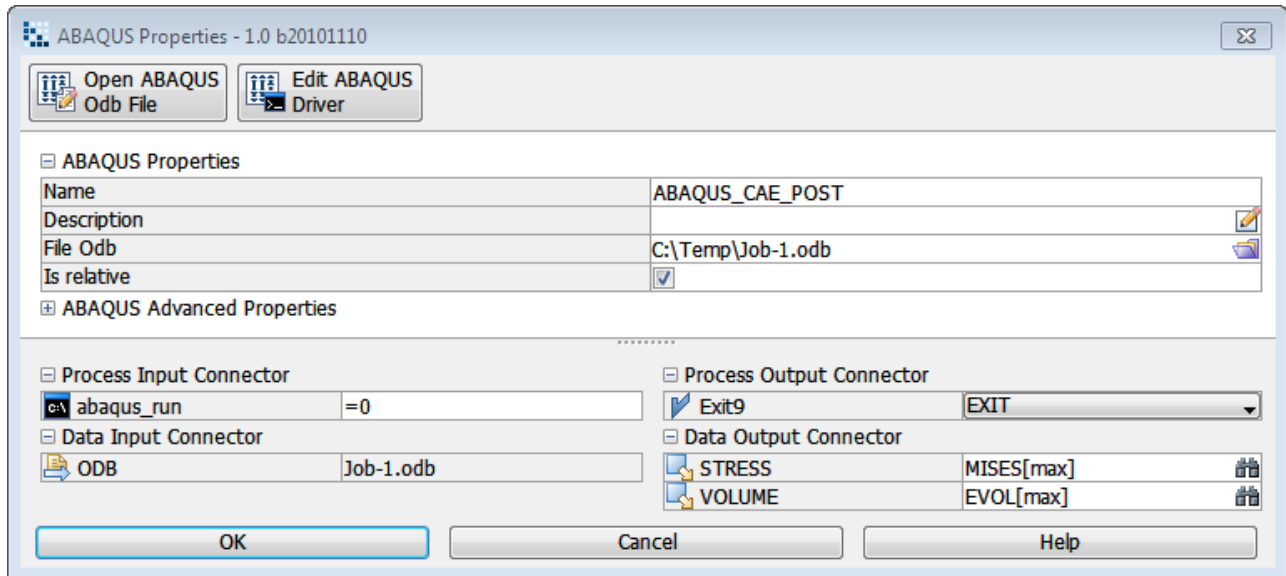


Figure 8-7: Abaqus-node properties.

In this problem, the volume of the beam and the stress is retrieved from odb-file in Abaqus and has to be tagged and located in the Abaqus-node see Figure 8.8.

Where the EVOL represents the volume of the element, the element used during this problem is B22- 3 node quadratic beam element. It should be noted here that since the same beam cross section is used throughout a particular design and the mesh density is constant for all designs, EVOL is directly proportional to the total volume of the structure.

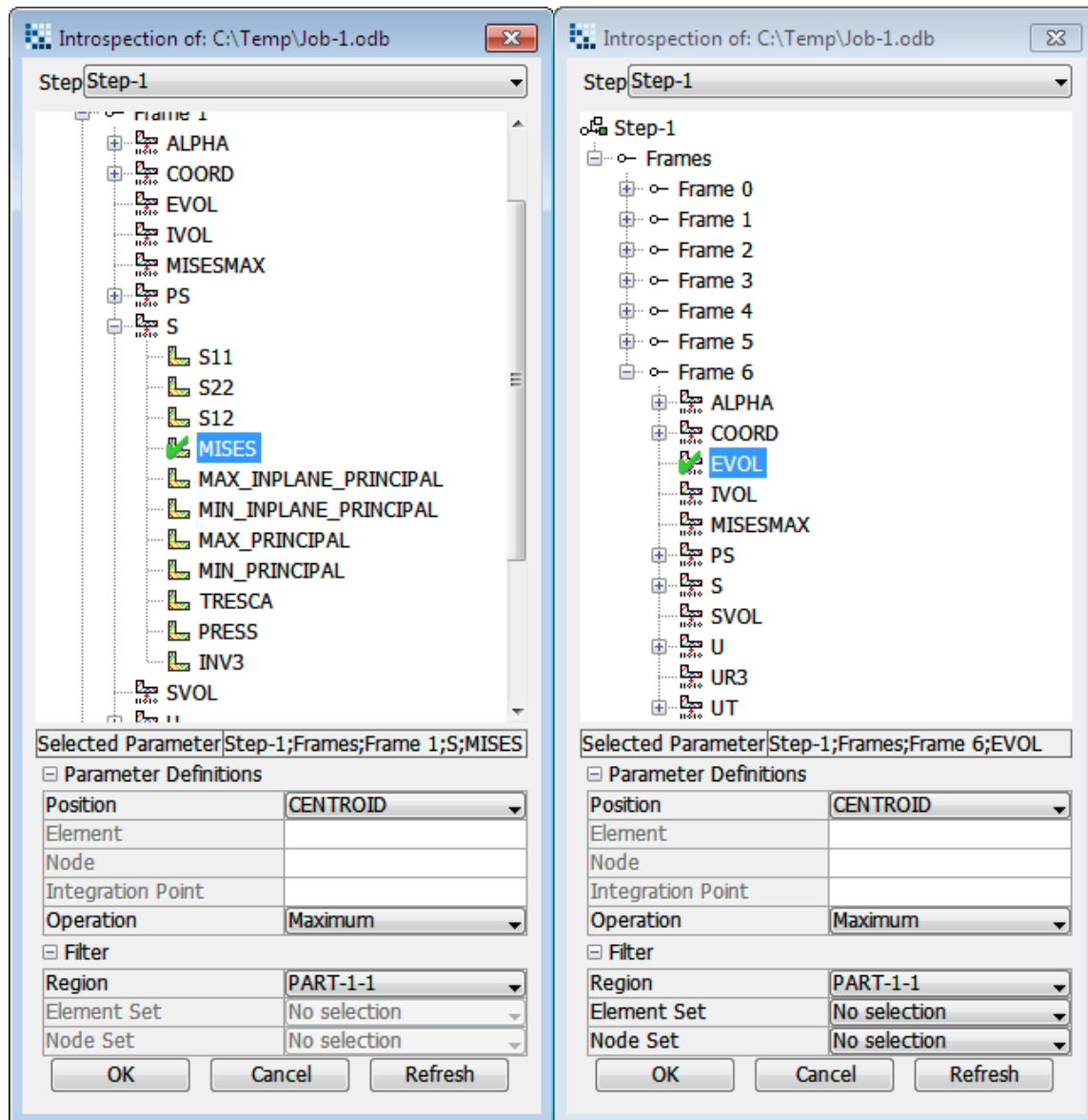



Figure 8-8: Odb-file Introspection at the Abaqus-node.

The output data is passed as input to the objective node  in which the objective function is written as a mathematical form. It is necessary to specify whether the objective should be minimized or maximized, see Figure 8.9. In this problem the workflow has two objectives which are minimize the volume and stress on the structure.

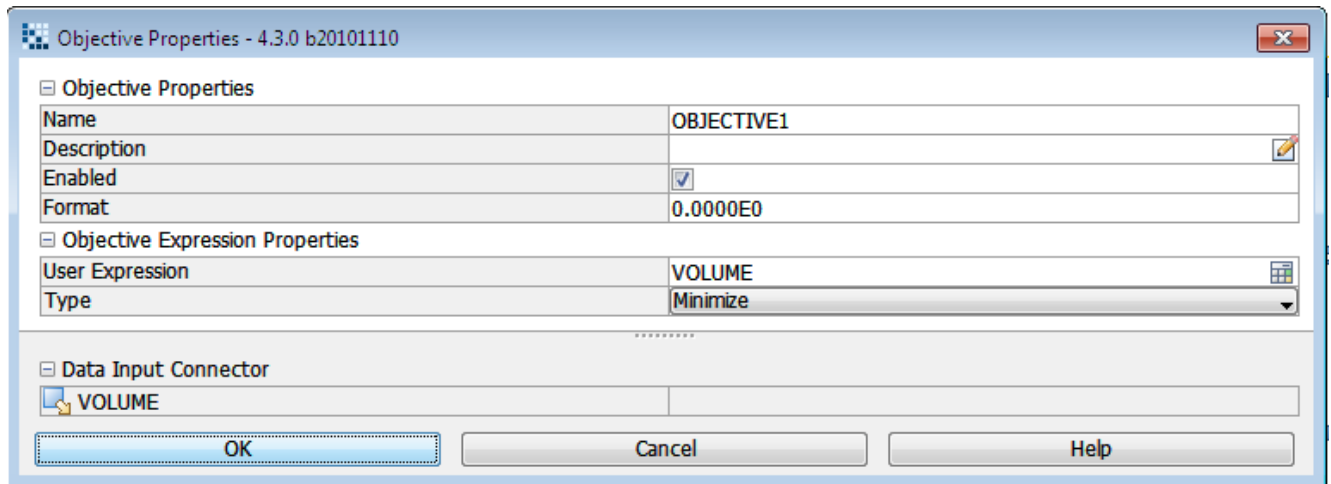


Figure 8-9: Objective-node properties.

Figure 8.10, presents the Pareto front result from this analysis.

The choice of optimal design from the Pareto front depends on the weight of the two objectives. If the most important objective is to minimize volume, design C is the best performing; this design has low volume but high stresses. If instead reduction of stress is the most important objective, design A should be retrieved. Figure 8.11, indicates the characteristics of these designs picked from different places in the Pareto front.

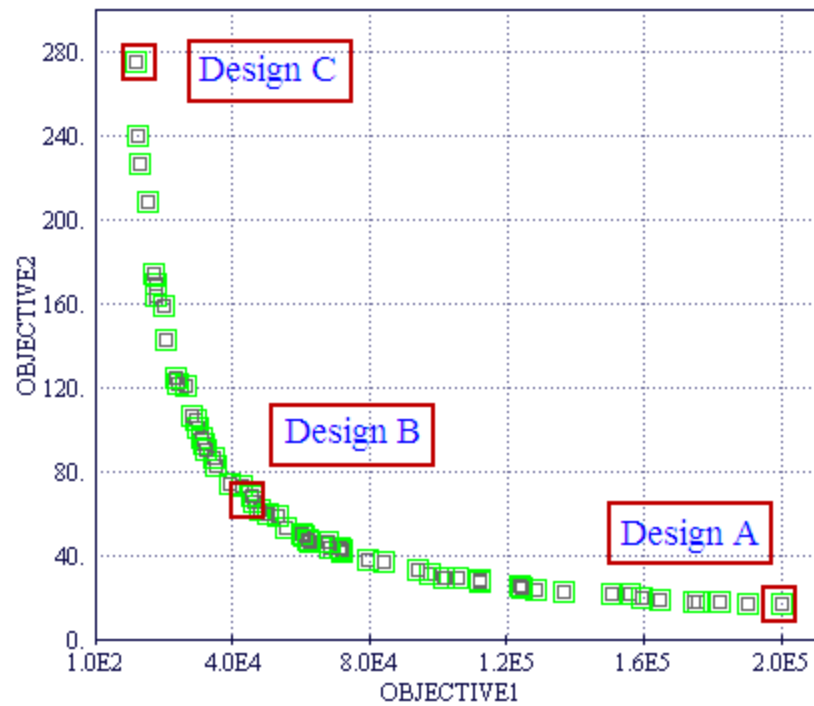


Figure 8-10: Pareto front of the two-bar truss problem.

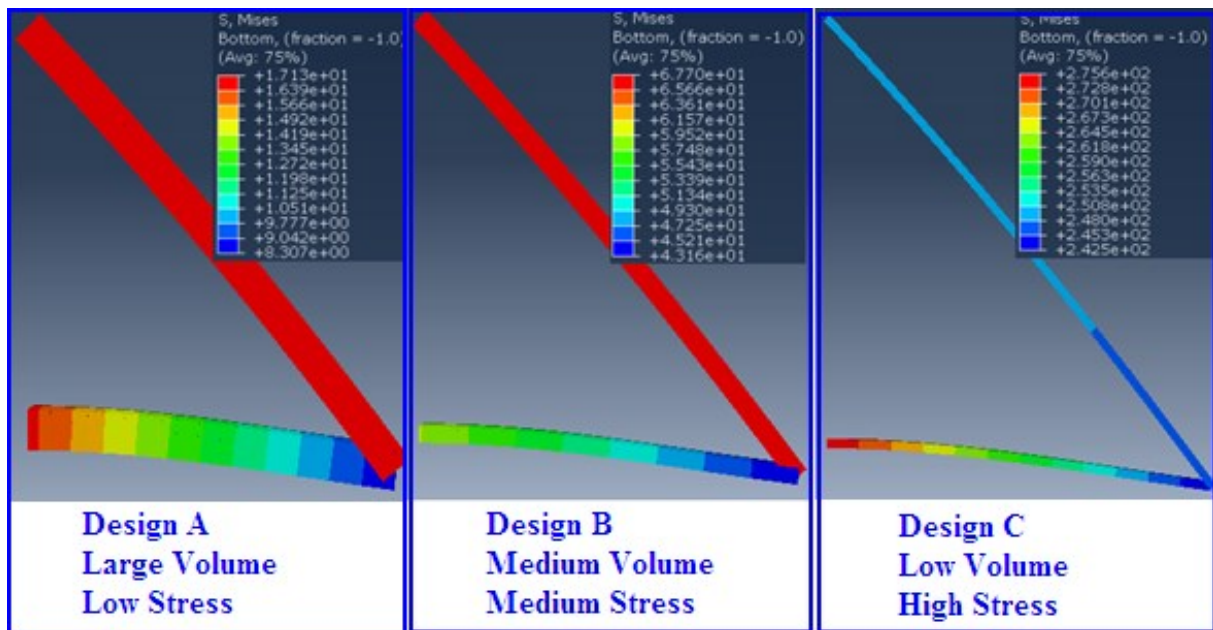


Figure 8-11: Three Pareto optimal solutions provided by modeFRONTIER.

8.2.2 Multi-objective Optimization of Plate with Hole

A plate model with a hole is as shown above in Figure 8.12. It has length of 1000 mm, width of 100 mm and thickness of 1 mm. The design variables in this model that are to be varied are the x-coordinate, and the radius of the hole (R). The plate is completely fixed at the left hand vertical side and a force is applied on the right hand vertical side. The plate structure is subjected to loads of ($F_x=10$ N and $F_y=10$ N) at right hand edge

The material properties of the plate is Young's Modulus of elasticity, $E= 210 \cdot 10^3$ N/mm², Poisson's ratio, $\nu = 0.3$

The objective of this analysis is to find the optimal design variable values so that at the free end the displacement in the x-direction is minimized, and the displacement in the y-direction is maximized

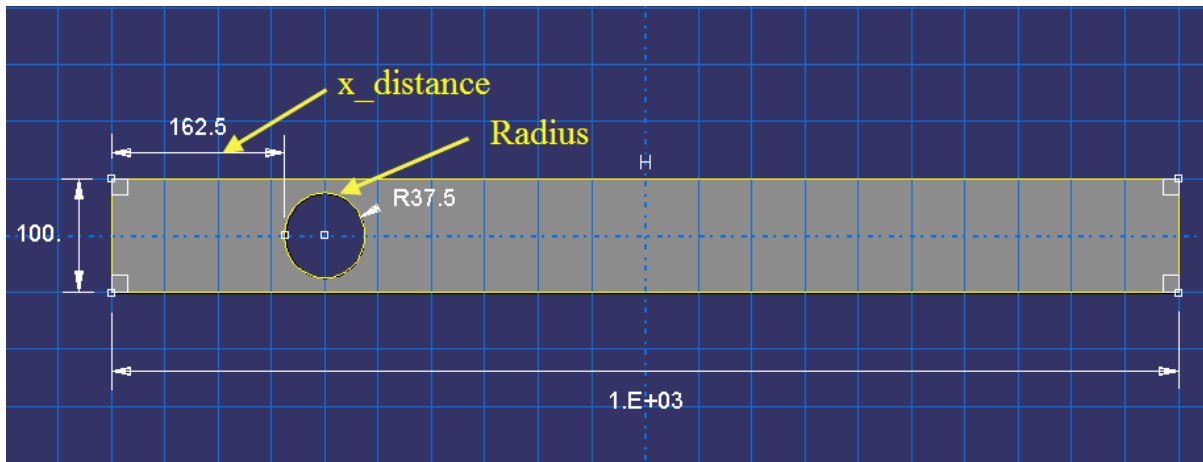


Figure 8-12: Dimension of the plate.

1- Problem Formulation

The optimization analysis of the plate structure can be formulated as follows

$$\text{Objective 1 } \mathbf{Min_X_DISP} \quad (8.3)$$



$$\text{Objective 2 } \mathbf{Max_Y_DISP} \quad (8.4)$$





Variables bound

$$12.5 \leq \mathbf{Radius} \leq 37.5 \quad [\text{mm}]$$

$$100 \leq \mathbf{x_distance} \leq 300 \quad [\text{mm}]$$

1 - Description of the Workflow

Figure 8.13, indicates the workflow set-up in modeFRONTIER 4.3.0.  **Radius** and  **x_distance** are the input variable nodes. The upper and lower limit of these design variables is set while configuring this node in modeFRONTIER. The procedure used in modeFRONTIER for this problem is the same as used in the previous problem.

The output variables x-displacement and y-displacement are contained in the **X_DISP**  and the **Y_DISP**  nodes and the objective of minimizing x-displacement and maximizing y-displacement is specified by objective nodes **Min_X_DISP**  and **Max_Y_DISP** .

For DoE in this problem Sobol has been used, with FMOGA-II optimization algorithm

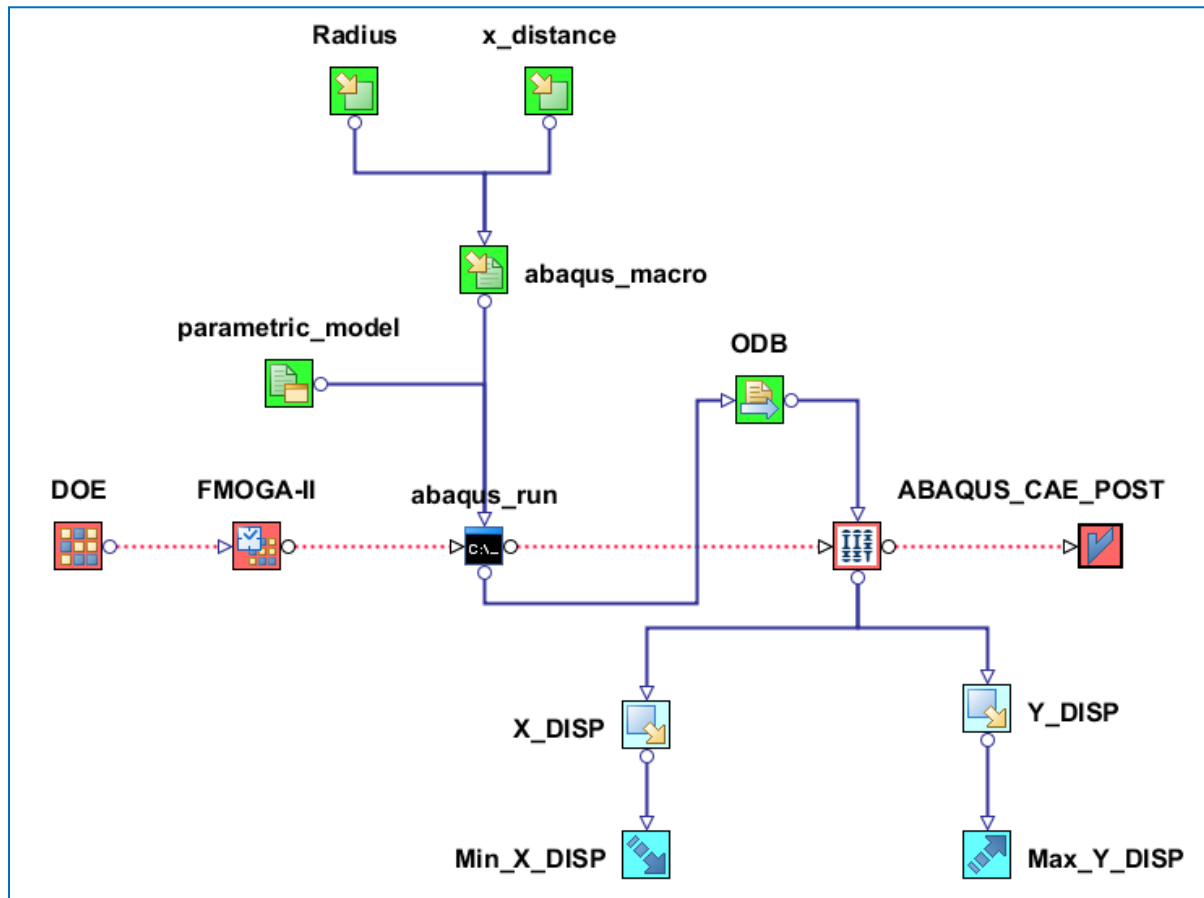


Figure 8-13: modeFRONTIER workflow for plate with hole problem

2 - Problem Results and Discussion

To investigate the effect of mesh density on the optimization result, three different cases have been studied in this section, Case 1 (1117 elements), Case 2 (2253 elements) and Case 3 (3113 elements), see Figure 8.14.

The Fast Multi-Objective Genetic Algorithm (FMOGA-II) has been used, to obtain the Pareto frontier (set of non-dominated design) starting with a Sobol sequence as an initial population. The initial population has been made to evolve for 25 generations trying to minimize the tip displacement in x direction and maximize this displacement in y direction.

From Figures 8.15 to 8.17 and Table 8.1, it is clear that an increase in the mesh density will improve the Pareto front distribution, additionally an increase in mesh density will increase the number of designs on the Pareto front from 87 in case 1 to 106 in case 3. However, there is no significant change in the shape or position of the Pareto front and hence no significant change in performance at points along this front.

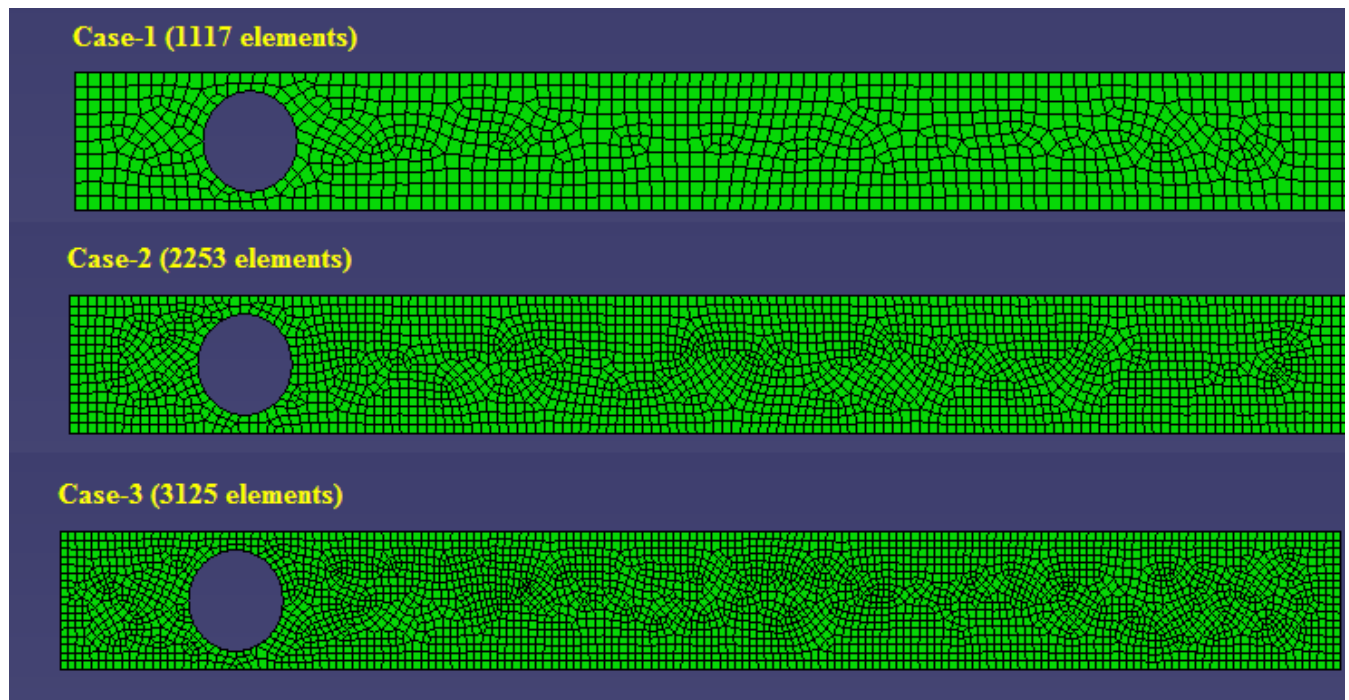


Figure 8-14: Three- case study according to the mesh size

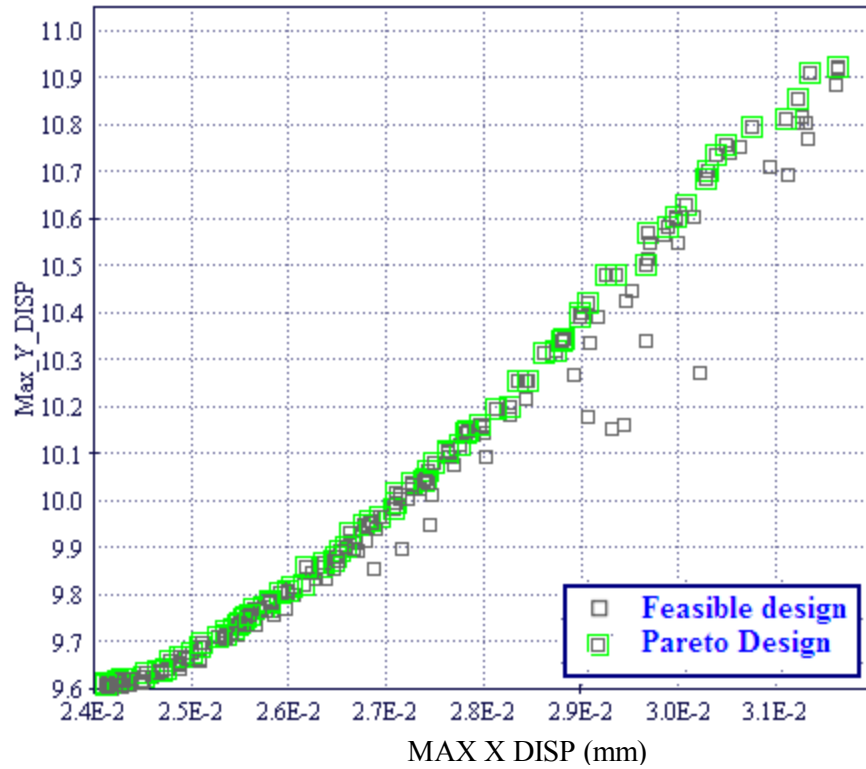


Figure 8-15: Pareto frontier of case 1 with (1117 elements)

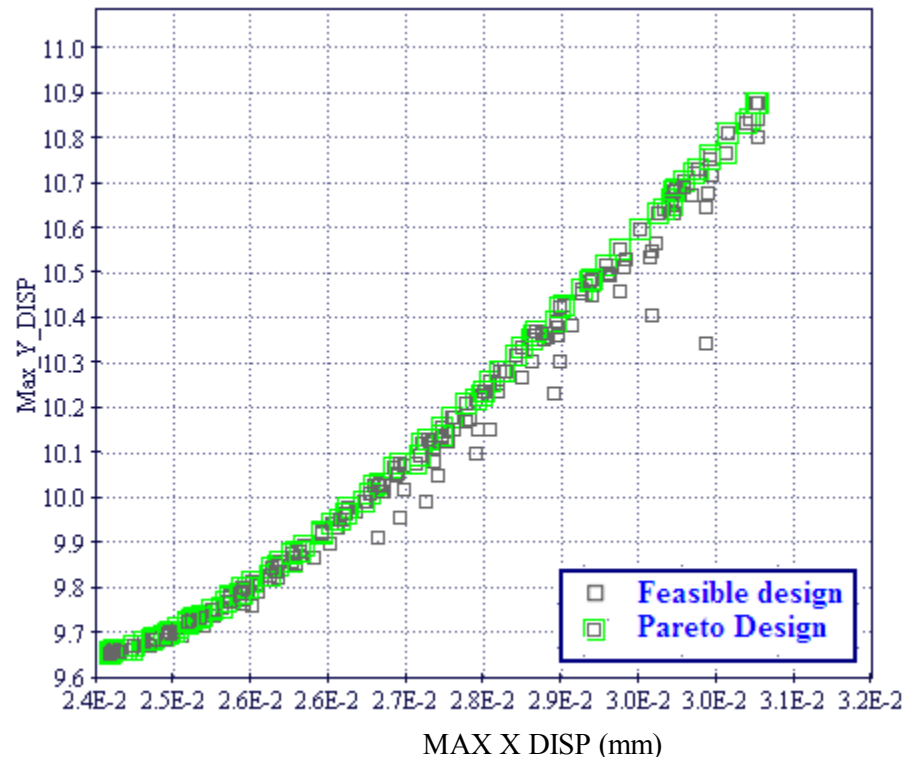


Figure 8-16: Pareto frontier of case 2 with (2253 elements)

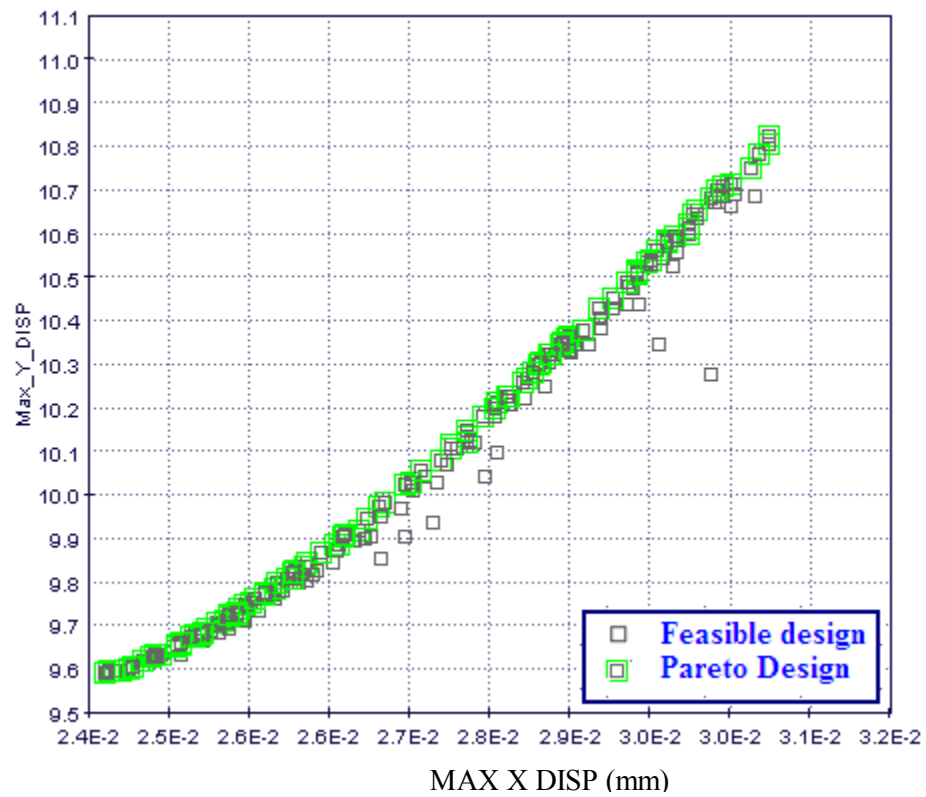


Figure 8-17: Pareto frontier of case 3 with (3113 elements).

8.3 Traditional Optimization of Welded Joints

In this section, the multi-objective optimization of a welded joint is presented. The problem is a plane stress problem of a welded joint as shown in Figure 8.18. Details of the welded joint geometry and design parameters are presented in Chapter 6.

Since the problem is regarding high cycle fatigue, the analysis will be linear and does not take into account any plastic behavior of the material.

The material properties of the joint (Steel S355JR) are: Young's Modulus of elasticity, $E = 207 \times 10^3 \text{ N/mm}^2$, Poisson's ratio, $\nu = 0.3$, Yield strength, $\sigma_y = 372 \text{ MPa}$

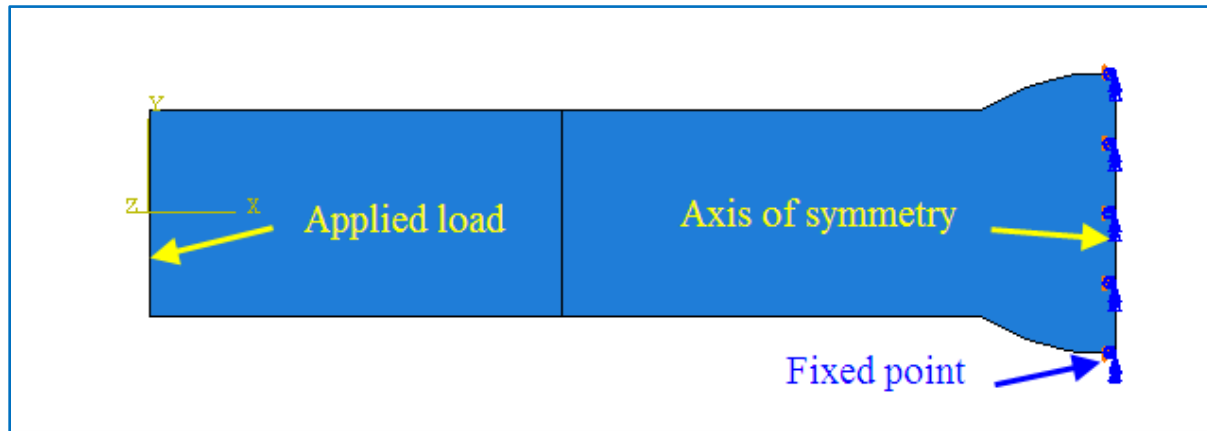


Figure 8-18: Geometry of welded joint.

8.3.1 Problem Formulation

The optimization task is to find the design parameters which minimize the stress under tensile and bending loads whilst constraining the maximum stress to be less than or equal to the yield stress of the material.

The multi-objective optimization problem can be formulated in the following form:

Objectives:

$$\text{Minimize: } f_1 = \sigma_1 \quad (\text{maximum Von-Misses stress under tensile load}) \quad (8.5)$$

$$\text{Minimize: } f_2 = \sigma_2 \quad (\text{maximum Von-Misses stress under bending load}) \quad (8.6)$$

Constraints:

$$\sigma_1 \leq \sigma_y$$

$$\sigma_2 \leq \sigma_y$$

Design variables:

$$0.5 \leq \rho_u \leq 2 \quad [\text{mm}]$$

$$0.5 \leq \rho_l \leq 2 \quad [\text{mm}]$$

$$12 \leq \alpha_u \leq 25 \quad [^\circ]$$

$$20 \leq \alpha_l \leq 30 \quad [^\circ]$$

$$1.2 \leq h_u \leq 2 \quad [\text{mm}]$$

$$1 \leq h_l \leq 1.2 \quad [\text{mm}]$$

Where ρ_u , ρ_l , α_u , α_l , h_u , h_l are the upper weld toe radius, lower weld toe radius, upper weld toe angle, lower weld toe angle, upper reinforcement and lower reinforcement respectively

8.3.2 Description of the Workflow

Figure 8.19, illustrates the workflow set-up of welded joint problem in modeFRONTIER

The numbered corresponding nodes are:

- 1- Input variables.
- 2- Python script and CAE model (as described in section 6.3.3 and appendix B).
- 3- DoE and Scheduler: FMOGA-II has been used with an initial population defined using a Sobol set.
- 4- Transfer file and DOS Batch script.
- 5- Abaqus post processing Node
- 6- Output variables.
- 7- Objective functions.
- 8- Constraints.

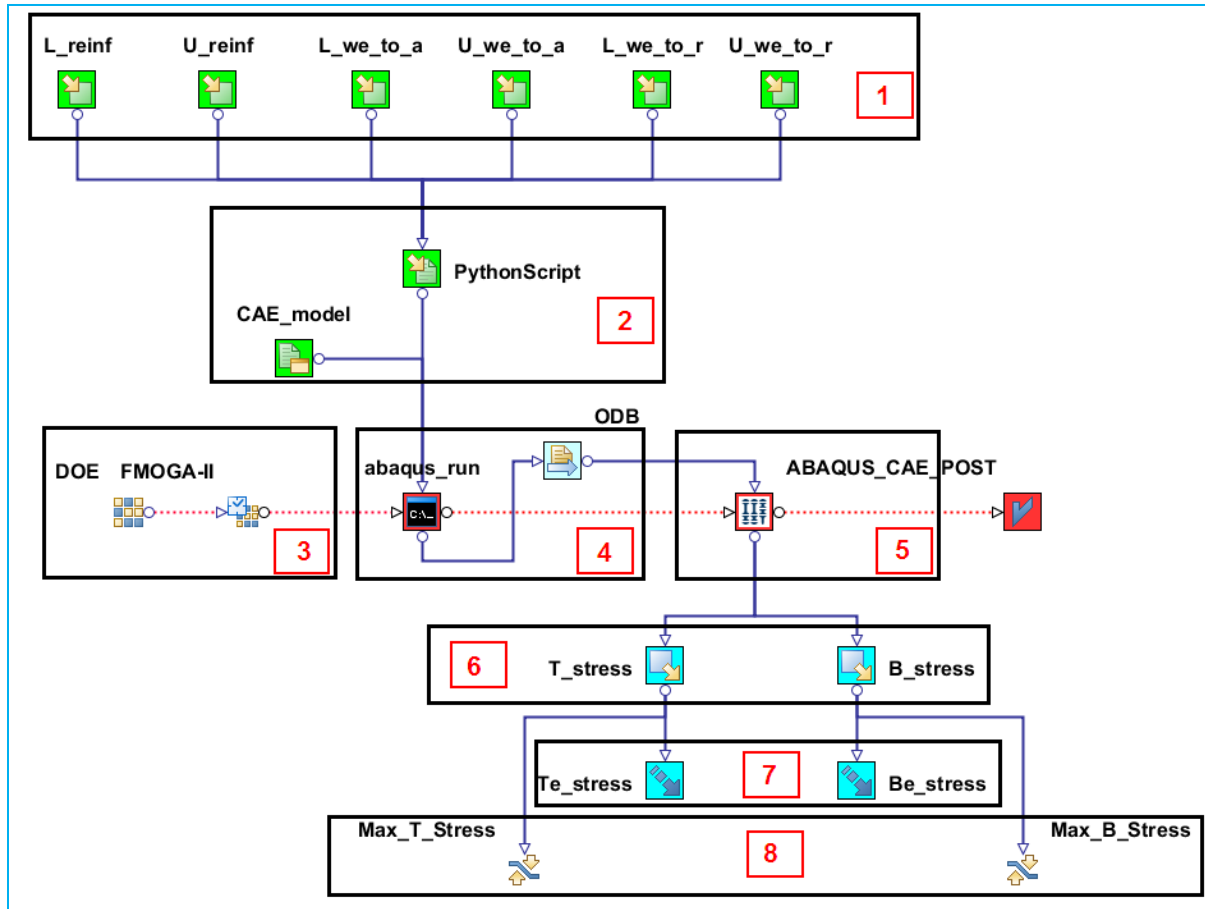
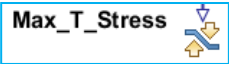



Figure 8-19: Workflow modeFRONTIER of welded joint problem.

Due to naming restriction in modeFRONTIER, the input and output parameters have been named as shown in Table 8.1.

Table 8-1: Input and output variables name in modeFRONTIER software

Variables name	Variables name in modeFRONTIER
Upper weld toe radius	U_we_to_r
Lower weld toe radius	L_we_to_r
Upper weld toe angle	U_we_to_a
Lower weld toe angle	L_we_to_a
Upper reinforcement	U_reinf
Lower reinforcement	L_reinf
Bending stress	Be_stress
Tensile stress	Te_stress

Node 8 has not been used in previous analysis. The purpose of this node ( , ) is to keep the Von-Mises stress in tensile and bending loads under a limiting value. The setting of this constraint node is indicated in Figure 8.20.

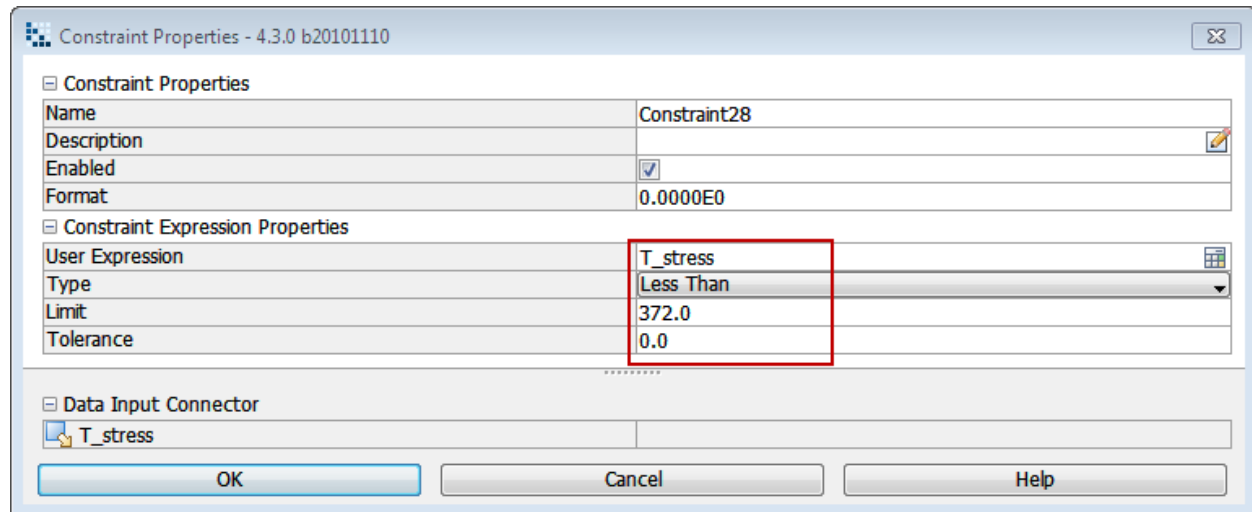


Figure 8-20: Constraint node setting.

8.3.3 Performing Optimization

Table 8.2 summarizes the optimization run. The feasible designs are shown in the tensile stress vs. bending stress in white Figure 8.21, while the non-feasible ones are in yellow.

Table 8.3, shows the initial ranges of the design parameters alongside these ranges for the Pareto optimal solutions.

The designs in Figure 8.21 can be divided into three groups feasible, non-feasible and Pareto designs. There are many non-feasible designs because they break the constraints, additionally the Pareto front is very small because the two objective functions are not conflicting.

From the results in Table 8.3, seems that there is a tight tolerance on the weld toe radius of the lower half of the joints. However, it should be noted that this small range of values is towards one end of the range of values allowed for the full population. In other words, the lower weld toe radius should be as large as possible within this range of possible values. Conversely, the range of Pareto optimal values for the upper weld toe radius covers almost the entire range possible for this variable.

Table 8-2: Design parameters problem

Parameter	Value
Initial population	50 Sobol
Generations	20 FMOGA-II
Total designs	1000
Error and non-feasible designs	0
Pareto design	11

Table 8-3: Initial and optimized design parameters

Design Parameters	Initial value		Optimized value		Objective1 (Tensile stress) (MPa)	Objective2 (Bending stress) (MPa)
	Min	Max	Min	Max		
Upper weld toe radius	0.5 [mm]	2	0.527	1.933	322.5-342.2	292.3-307.4
Lower weld toe radius	0.5 [mm]	2	1.9	2		
Upper weld toe angle	12 [°]	25	12	22.26		
Lower weld toe angle	20 [°]	30	20	23		
Upper reinforcement	1.2 [mm]	2	1.2	1.92		
Lower reinforcement	1 [mm]	1.2	1.1	1.2		

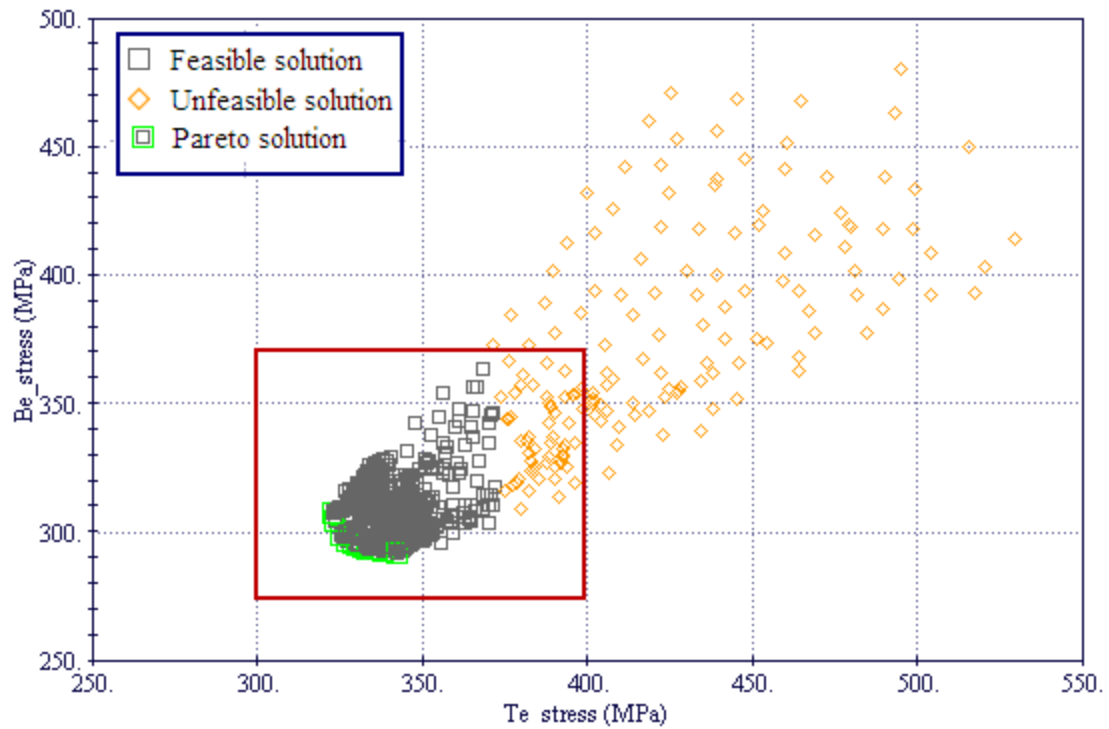


Figure 8.21: Tensile stress vs. bending stress

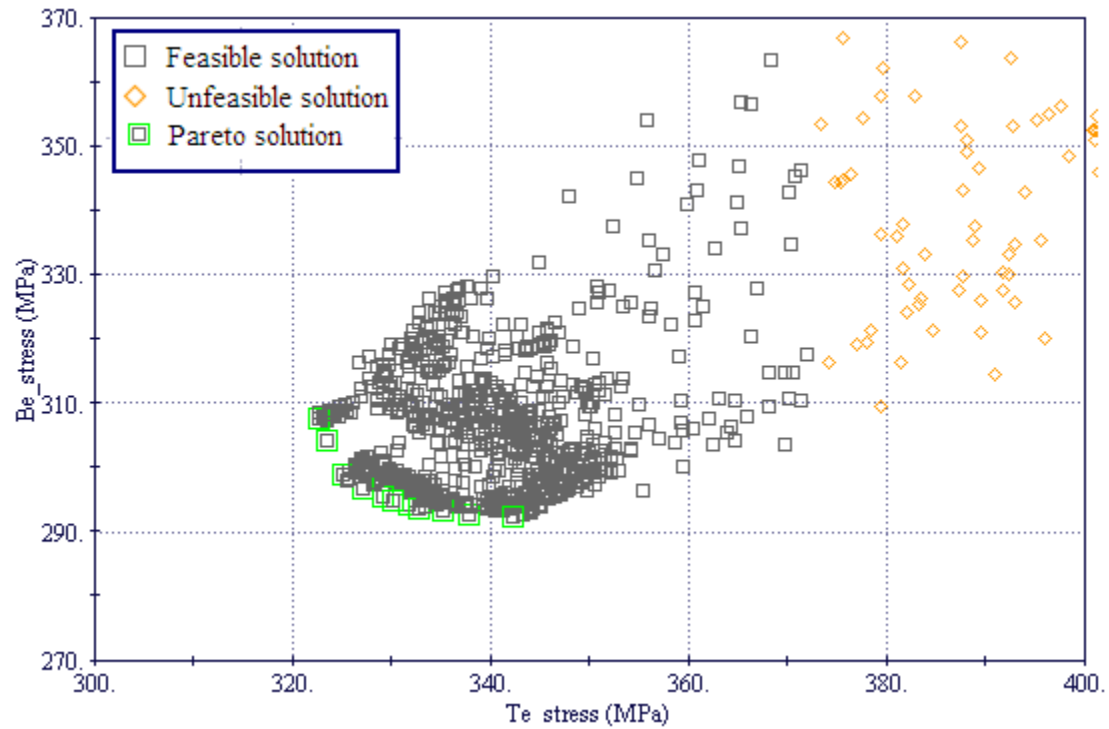


Figure 8-21: Tensile stress vs. bending stress.

8.4 Robust Optimization of Welded Joints

From traditional optimization in the previous section, the construction of a mathematical function which describes the Pareto front is performed by using Matlab software. From this equation, it is easy to describe the constraint on the Pareto front in order to perform a robust optimization as described in detail in Section 5.2.1.

The equation of Pareto is presented in Figure 8.21, the form of this equation is:

$$f(x) = ax^b \quad (8.7)$$

Where $a = 2.566 \text{ E}+4$ and $b = -0.7681$

In the same way and from equation 8.7, the constraint with different level can be written in Table 8.4 and presented in Figure 8.22.

Table 8-4: Pareto front with constraint

Tolerances	Equation
5%	$f(x) = ax^{-0.7596}$
-5%	$f(x) = ax^{-0.7777}$
a	2.556E+4

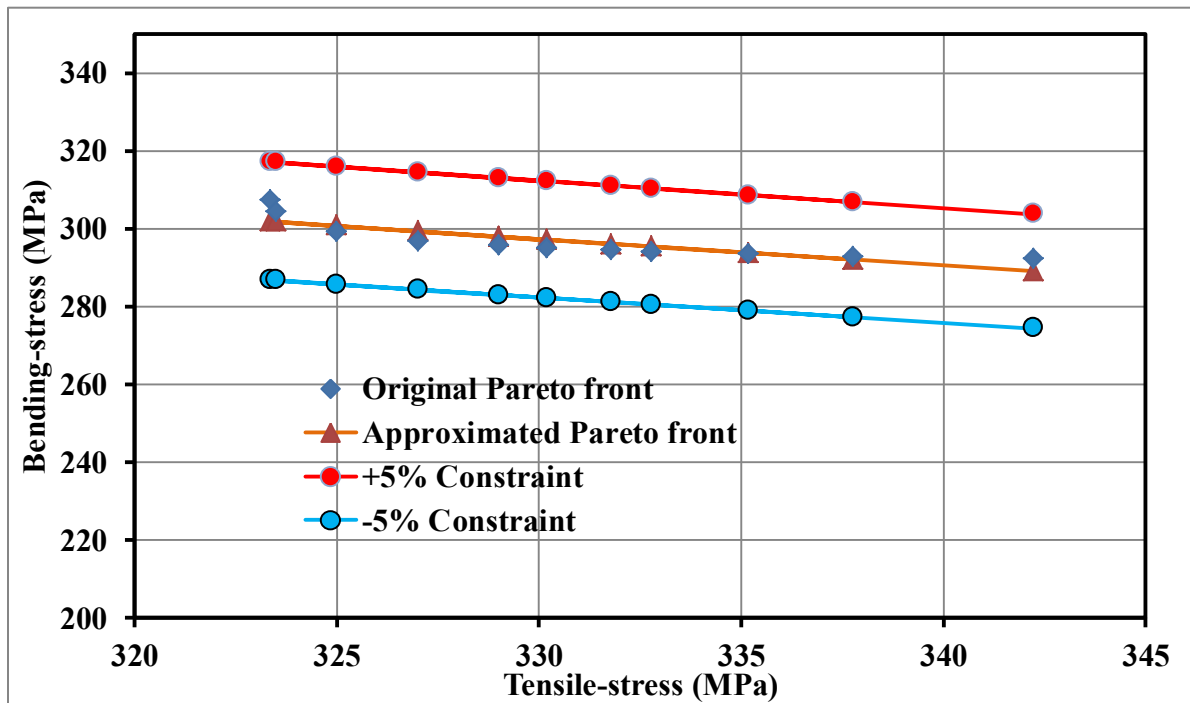


Figure 8-22: Representation of Pareto front with different levels of constraints.

8.4.1 The Problem Definition

The simple work flow for a multi-objective robust design optimization of welded joint is shown in the Figure 8.23, the detailed description of robust optimization in modeFRONTIER have been conducted in Chapter 5.

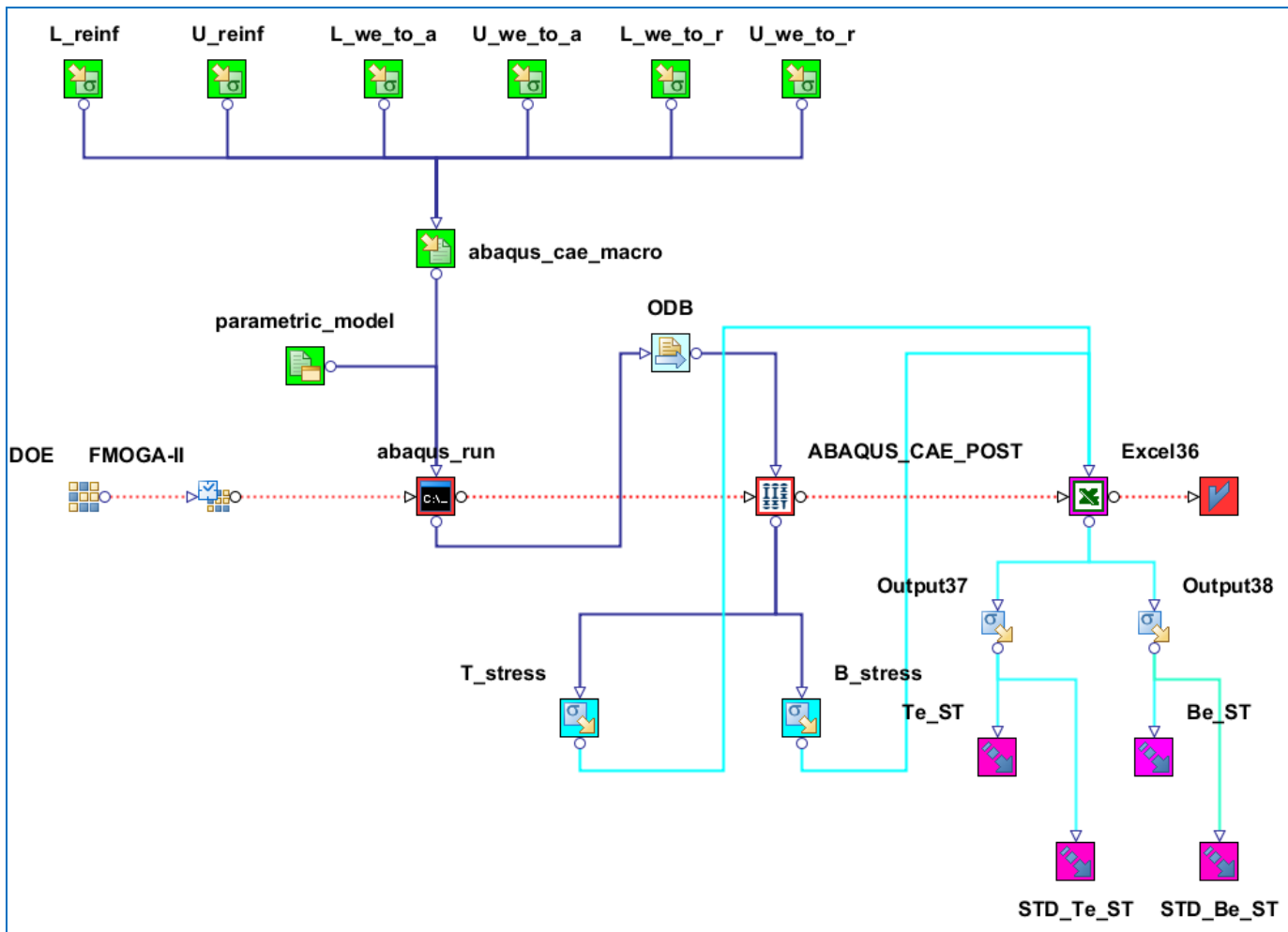



Figure 8-23: The multi-objective robust design optimization work flow.

All the procedures and nodes are the same as in the traditional optimisation except the Excel node () which was used in this case to defining the constraint on the Pareto front. Equation 8.7 was used inside the Excel node with the specified constraint.

The second difference with the traditional optimization is the scheduler node where the number of subsidiary points around a design point is defined. The modeFRONTIER will run 50 designs using a normal distribution around each nominal design point, as shown in Figure 8.24.

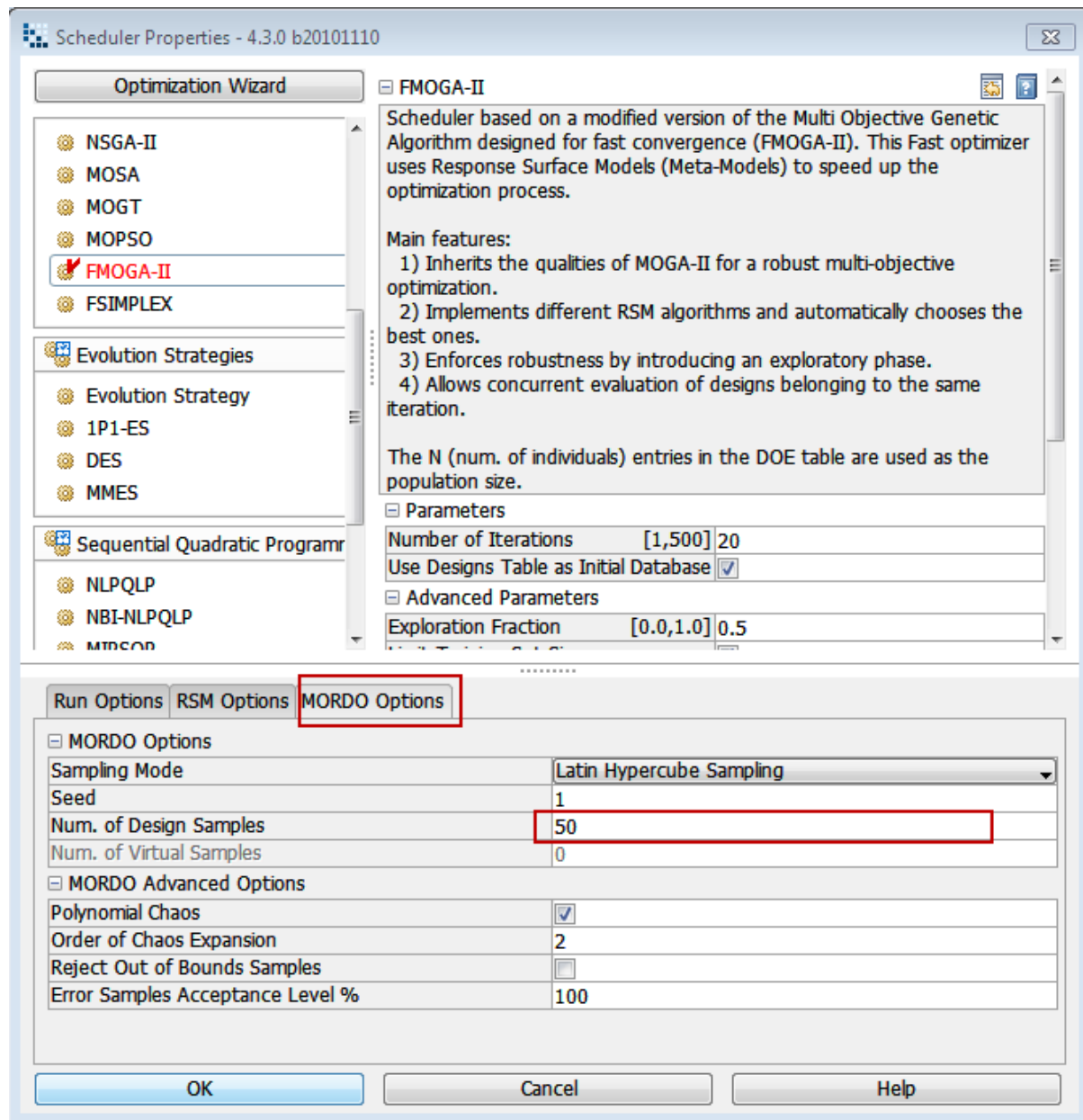


Figure 8-24: The scheduler node properties.

8.4.2 Performing Optimization

After final adjustment of the input design variables by defining the standard deviation and nominal values of these variables, the optimization problem was setup. The Fast Multi-Objective Genetic Algorithm (FMOGA-II) was selected and the process ran for 20 iterations with a 50 point Sobol sequence as the initial population. This resulted in 50,000 separate designs being evaluated (50 Sobol initial population * 20 generations * 50 normally distributed points).

Three different cases according to the value of standard deviation of input variables have been introduced, with a +5%, -5% constraint on the Pareto front. The input design variable values are presented in Table 8.5. The goal of the optimization was to find the minimum standard deviation of the two objective functions.

For all three cases the number of points in the Pareto set was very small. The values of the design parameters along with the peak stresses in tension and bending are shown in Table 8.6.

In order to better understand the distribution of design parameters relating to the minimum standard deviation to the values have been normalized before being plotting the radar chart. The normalized parameters are indicated in Table 8.6

The distribution of the optimized input variables between upper and lower limits are presented in radar charts (Figures 8.25 to 8.27). From these figures it is clear to see that the distribution of lower and upper reinforcement is categorized in two patterns, the first one with a low value of upper reinforcement and a high value of lower reinforcement (case-1), whilst the second pattern contains low and moderate values of lower reinforcement and extremely high values of upper reinforcement (case-2 and case -3).

Regarding the lower and upper weld toe radii, these results indicate clearly the existence of two patterns as well. The first pattern has a moderate upper weld toe angle and higher value of lower weld toe angle (case-1). The second pattern represents a moderate value of lower weld toe angle and an extremely high value of the upper weld toe angle (case-2 and case -3).

In the distribution of lower weld toe angle, the distribution of these variables takes the one pattern with moderate to high values (case-1, 2 and 3).

Regarding the distribution of upper weld toe angle, it is clear that the distribution of this design parameters divided into two patterns, the first one with moderate high value (case-1), whilst the second one with moderate to low values of this variable (case-2 and case-3).

Table 8-5: Input design parameters of three cases

Design parameters for Case-1			
Input Variable Name	Lower bound	Upper bound	Standard deviation
Upper weld toe radius	0.5	5	0.1
Lower weld toe radius	0.5	5	0.1
Upper weld toe angle	12	25	0.95
Lower weld toe angle	20	30	0.95
Upper reinforcement	1	3	0.125
Lower reinforcement	1	3	0.125
Output design parameters			
Total designs	Feasible designs	Error designs	Pareto designs
1000		593	3
Design parameters for Case-2			
Input Variable Name	Lower bound	Upper bound	Standard deviation
Upper weld toe radius	0.5	5	0.05
Lower weld toe radius	0.5	5	0.05
Upper weld toe angle	12	25	0.475
Lower weld toe angle	20	30	0.475
Upper reinforcement	1	3	0.0625
Lower reinforcement	1	3	0.0625
Output design parameters			
Total designs	Feasible designs	Error designs	Pareto designs
1000		429	3
Design parameters for Case-3			
Input Variable Name	Lower bound	Upper bound	Standard deviation
Upper weld toe radius	0.5	5	0.025
Lower weld toe radius	0.5	5	0.025
Upper weld toe angle	12	25	0.237
Lower weld toe angle	20	30	0.237
Upper reinforcement	1	3	0.03125
Lower reinforcement	1	3	0.03125
Output design parameters			
Total designs	Feasible designs	Error designs	Pareto designs
1000		511	2

Table 8-6: Robust design parameters of three cases

Input design parameters corresponding robust design for Case-1								
Id	L_reinf	U_reinf	L_we_to_r	U_we_to_r	L_we_to_a	U_we_to_a	B_stress	T_stress
1	2.39	1.48	4.64	2.21	26.61	21.38	292.53	342.42
2	2.30	1.38	4.72	2.24	26.39	21.50	295.38	346.21
3	2.61	1.38	4.72	2.71	29.26	22.15	284.44	334.26
Normalized values of design parameters								
Id	L_reinf	U_reinf	L_we_to_r	U_we_to_r	L_we_to_a	U_we_to_a	B_stress	T_stress
1	0.694	0.241	0.920	0.380	0.661	0.722	292.53	342.42
2	0.648	0.191	0.938	0.387	0.639	0.731	295.38	346.21
3	0.804	0.188	0.938	0.491	0.926	0.781	284.44	334.26
Input design parameters corresponding robust design for Case-2								
Id	L_reinf	U_reinf	L_we_to_r	U_we_to_r	L_we_to_a	U_we_to_a	B_stress	T_stress
1	1.48	2.96	2.3	4.92	27.8	16	302.98	335.8
2	1.45	2.91	2.47	4.88	27.77	16.78	300.04	332.36
3	1.51	2.87	2.47	4.8	27.81	16.72	299.39	331.69
Normalized values of design parameters								
Id	L_reinf	U_reinf	L_we_to_r	U_we_to_r	U_we_to_a	L_we_to_a	B_stress	T_stress
1	0.240	0.980	0.400	0.982	0.780	0.308	302.98	335.80
2	0.225	0.955	0.438	0.973	0.777	0.368	300.04	332.36
3	0.255	0.935	0.438	0.956	0.781	0.363	299.39	331.69
Input design parameters corresponding robust design for Case-3								
Id	L_reinf	U_reinf	L_we_to_r	U_we_to_r	L_we_to_a	U_we_to_a	B_stress	T_stress
1	1.7	2.9	2.31	4.88	27.958	17.2	302.723	334.304
2	1.6	2.859	2.525	4.78	26.266	15.51	294.519	314.182
Normalized values of design parameters								
Id	L_reinf	U_reinf	L_we_to_r	U_we_to_r	U_we_to_a	L_we_to_a	B_stress	T_stress
1	0.350	0.950	0.402	0.973	0.796	0.400	302.723	334.304
2	0.300	0.930	0.450	0.951	0.627	0.270	294.519	314.182

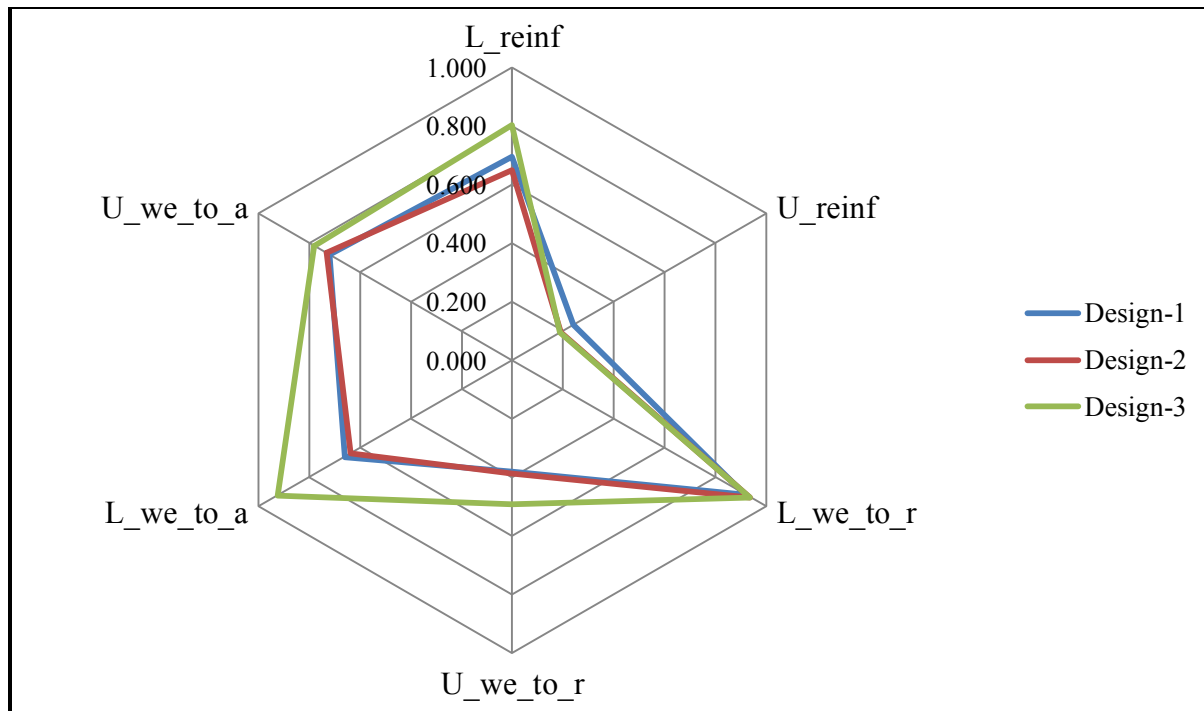


Figure 8-25: Radar chart for Case-1

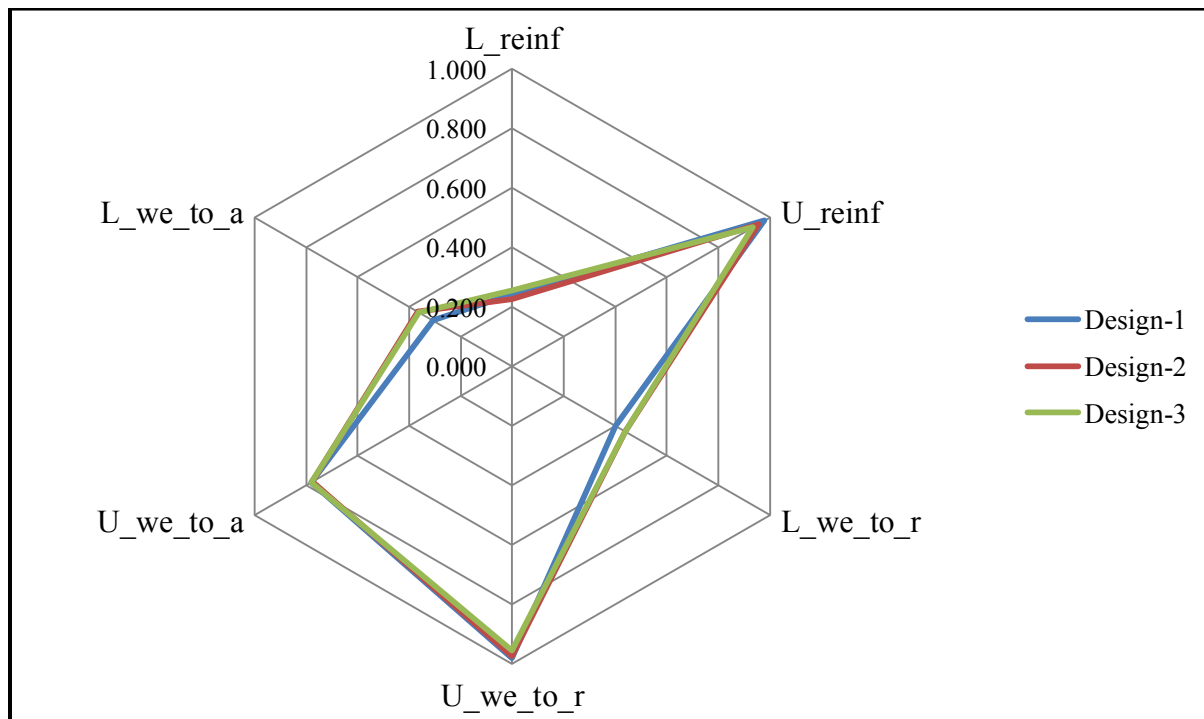


Figure 8-26: Radar chart for Case-2

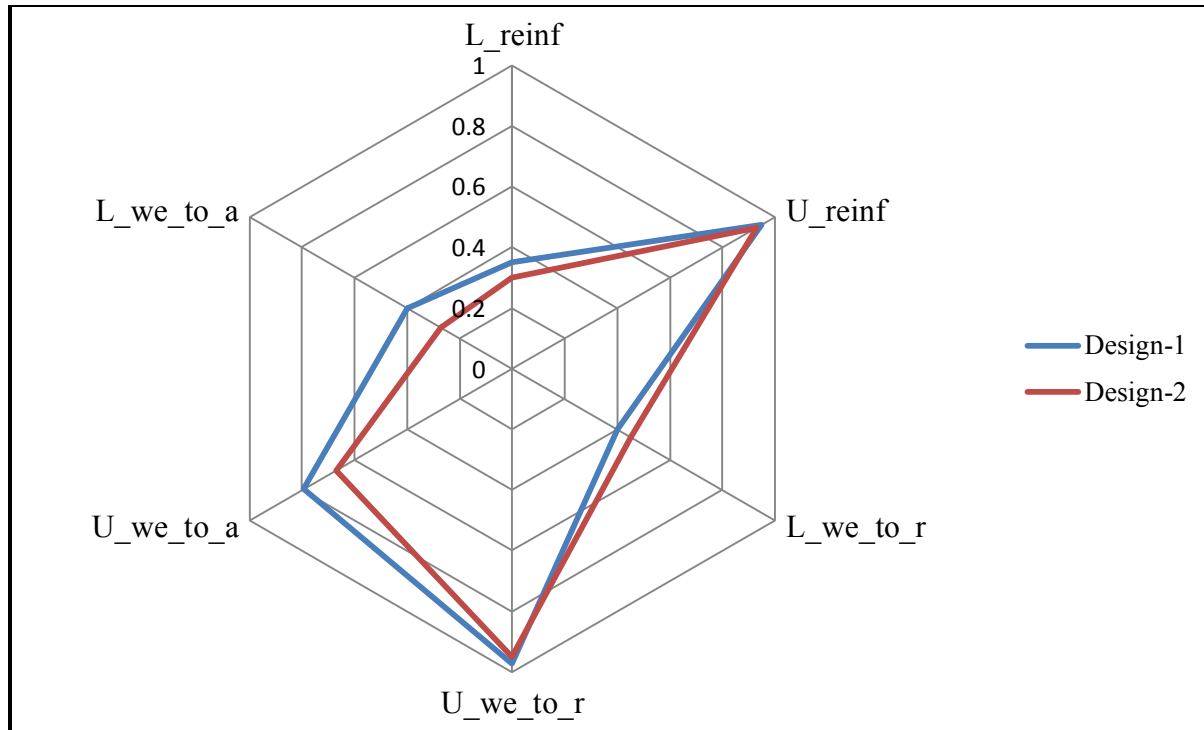


Figure 8-27: Radar chart for Case-3

8.5 Conclusion

Two multi-criteria examples have been used to demonstrate that the ABAQUS FEA software can be integrated with the modeFRONTIER optimization software to allow optimization of structures with complex geometries to be undertaken.

For the plate with a hole example increasing the mesh density improved the spread of Pareto optimal solutions.

Optimization of a welded joint with the objective of minimizing the peak stress in bending and tension resulted in a very small Pareto front. This was due to the two objectives not being in conflict. Additionally the Pareto optimal designs were at the extreme values of some geometrical parameters such as lower weld toe radius.

Robust optimization of the welded joint based on the Pareto front from traditional optimization with constraint of (+5%, -5%) produced a small number of designs with very similar values for the design parameters. i.e. out of the Pareto set, a single design could be selected as the most robust.

The values of the design parameters generating the robust design were dependent upon the standard deviation of the design parameters. However, as the standard deviation was reduced, the geometry of the robust weld converged to a single design.

Chapter 9



Conclusions and Recommendations for Further Work

Chapter nine: Conclusions and recommendations for further work

9.1 Introduction

As mentioned before the aim of this work was to produce a methodology for the robust design of butt welded joints. The following items were achieved during this study:

- The current state of the art regarding weld geometry optimization was identified.
- An appropriate tool for multi-criteria structural optimization was identified.
- A method of embedding finite element analysis within an optimisation tool was implemented.
- A method of identifying robust designs from within the Pareto set for multi-criteria problems was identified.

9.1.1 Main conclusions

The following major conclusions can be drawn from the results presented in this thesis.

- From the literature review it was apparent that the majority of the research on welded joints either deals with optimisation of the welding process (such as current, voltage) or considers one or two geometric parameters. However, there is a lack of knowledge in the studies of all geometrical parameters and their effect on welded joints under combined load.
- The geometrical parameters of welded joints (weld toe radius, weld toe angle, weld reinforcement, and weld width) play a very important role in the service life or load capacity of welded joints.
- The results presented here showed that the Circumscription Metric (CM) gives a better indication of diversity in the solution set than the Pair Wise metric (PW), particularly for optimisation carried out using genetic algorithms (see Chapter 4 section 4.4..4)
- For all the problems analysed here, the Fast Multi-objective Genetic Algorithm (FMOGA-II) outperformed the other algorithms.
- Increasing the initial population will improve the diversity of both the feasible set and the Pareto set due to the increased search space.

- The finite element model identified the weld toe as the peak stress raiser and hence the most likely initiation point for fatigue cracking. This was confirmed by the practical tests.
- The parametric finite element model could predict stress concentrations over a wide range of weld geometries. Increasing the weld toe radius reduced the stress concentration factor, whilst Increasing the weld toe angle increased the stress concentration factor
- A method was established to ensure that robust design solutions were members of the Pareto set. The pattern of robust designs generated using this approach depends on the standard deviation of the input design parameters.
- The new design optimization methodology will aid the engineer in selecting from the Pareto set by significantly reducing the size of the solution set.

9.2 Recommendations for further work

In the work presented here a power function was used to approximate the Pareto front and the tolerance band around that Pareto front. However, this only provided a moderately good fit to the Pareto front in the welded joint analysis. Other techniques for approximating the Pareto front (for example spline curves) should be investigated.

The problems investigated here were two criteria problems. Hence, the Pareto front could be approximated with a line. Methods of approximating the Pareto front and defining a tolerance band around it in three or more dimensions will have to be developed for problems with more than two criteria.

For the welded joint analysis, it was shown that the nature of the robust solution depended on the degree of spread of the design variables. Other optimization problems should be investigated to determine if this is generally applicable. Also, the apparent convergence of the solution as the variability of the design variables is reduced should be confirmed.

In some optimization problems the Pareto front may exist on a boundary between the feasible set and the non-feasible set. Hence, variation in the criteria resulting from variations in the design variables will make certain design points non-feasible rather than Pareto optimal. This will effectively move the Pareto front in criteria space. The impact that this has on the subsequent selection of a robust design from the Pareto set requires investigation.

The methodology presented here used a tolerance band placed around an approximation of the Pareto front to define the feasible set for the robust optimization analysis. However, with a sufficiently well populated Pareto front, it should be possible to determine a good approximation of the robust solution by determining the standard deviation of the design points within the current Pareto set. In addition to being well populated, the Pareto set would also have to be well distributed. Substantial further study would be required to determine the design point density and Pareto set uniformity required to generate a good approximation to the robust solution.

References

- ABAQUS (2008). "Getting Started with ABAQUS: Interactive Edition" Version 6.8. Dassault Systemes Simulia Corp., Providence, RI, USA
- ABAQUS (2010a). "ABAQUS Scripting User's Manual" Version 6.10. Dassault Systemes Simulia Corp., Providence, RI, USA
- ABAQUS (2010b). "ABAQUS Analysis User Manual Volume IV Part VI Element " Version 6.10. Dassault Systemes Simulia Corp., Providence, RI, USA
- AGARWAL S., (2011). *Failure and Lifetime Assessment of Welded Stainless Steel Structures via Finite Element Modeling and Variance Based Sensitivity Analysis Methods*. [Online]. Available: <http://www.optiy.de/download/Saurabh%20PPT.pdf> [Accessed 9 June 2012].
- AMERICAN WELDING SOCIETY. (2010). AWS A3.0M/A 3.0:2010 an American National Standard, Approved by American National Standard Institute, Standard Welding Term and Definitions
- ASTM E-1823, (2000). *Standard Terminology Relating to Fatigue and Fracture*. Vol. 03.01. West Conshohocken, PA: ASTM.
- BAGCHI, T. P. (1993). Taguchi Methods Explained: Practical Steps to Robust Design, Prentice Hall of India, New Delhi.
- BARSOM J. & ROLFE S. T. (1999). *Fatigue and Fracture Control in Structures: Application of Fracture Mechanics*. American Society for Testing and Materials (ASTM), U.S.A.
- BENYOUNIS K. Y. & OLABI A. G. (2008). *Optimization of different welding process using statistical and numerical approaches- a reference guide*. Advances in Engineering Software, 36, 483-496.
- BIRK L. & HARRIES S. (2003) Year. *OPTIMISTIC - Optimization in Marine Design*. In: 39th WEGEMT Summer School, Berlin, Germany.
- BOUKHAROUBA T., GILGERT J. & PLUVINAGE G. (1999). *Role of stress concentration in welded joint fatigue*. Technology Law and Insurance, 4, 207-217.
- BRANKE J., DEB K., MIETTINEN K. & SLOWINSKI R. (2008). *Multi-objective optimization interactive and evolutionary approaches*, Berlin, Springer-Verlag.
- BRATLEY P. & FOX B. (1988). ALGORITHM 659 Implementing Sobol's quasi random Sequence Generator. ACTM Transactions on Mathematical Software. 14, 88-100.
- BRITISH STANDARD INSTITUTION. (2004). BS EN 10025-2:2004 *Hot rolled products of structural steel*. London: BSI.
- BRITISH STANDARD INSTITUTION. (2009). BS EN ISO 6892-1:2009. *Metallic materials- Tensile testing*. London: BSI.
- BRITISH STANDARD INSTITUTION. (2013). BS EN ISO 9692-1:2013 *Welding and allied processes types of joint preparation*. London: BSI.
- CAIAZZO F., CURCIO F., DAURELIO G. & MEMOLA F. (2004). Ti6Al4V sheets lap and

- butt joints carried out by CO2 laser: mechanical and morphological characterization. *Journal of Materials Processing Technology*, 149, pp. 542-552.
- CARMICHAEL D. (ed.) (1981). *Structural Modeling and Optimization*, West Sussex: Chichester.
- CARY B. (ed.) (1979). *Modern Welding Technology*: Prentice-Hill.
- CERIT M., KOKUMER O., & GENEL K. (2010). Stress concentration effects of undercut defect and reinforcement metal in butt welded joint. *Engineering Failure Analysis* 17, 571-578.
- CHAPETTI M. D. & JAUREGUIZAHAR L. F. (2011). Estimating the fatigue behavior of welded joints. *Procedia Engineering*, 10, 959-964.
- CHARNES A. & COOPER W. W. (1959). Chance constrained programming. *Management Science*, Vol. 6, pp. 73-79.
- CHEN L. & BANNET D. J. (2010). Polynomial approximation for two stage stochastic programming with separable objective. *International journal of Operations Research and Information Systems*, 1, 75-88.
- COSTA J., FERREIRA J. A. & ABREU P. (2010). Fatigue behavior of butt welded joints in a high strength steel. *Procedia Engineering* 2, 697-705.
- COWAN G. R., DOUGLAS J & BOLTZMANN, A. H., (1964). Explosive bonding. US patent application 3137937.
- DAISUKE S. (2005). ARMOGA An efficient Multi-Objective Genetic Algorithm. *ESTECO Technical Report*.
- DE JONG K., (2006). *Evolutionary Computation: A Unified Approach*. MIT Press.
- DEB K. (ed.) (1995). *Optimisation Method for Engineering Design*, New Delhi: Prentice-Hall
- DEB K. (ed.) (2001). *Multi-objective Optimization using Evolutionary Algorithms*: John Wiley and Sons.
- DEB K. AGARWAL S., PARTAP A. & MEYARIVAN T. (2000). *A Fast Elitist Non Dominated Sorting Genetic Algorithm for Multi-Objective Optimization NSGA-II*. Indian Institute of Technology.
- DEB K., PARTAP A., AGARWAL S. & MEYARIVAN T. (2002). A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*. 6, pp. 182-197.
- DUFFIN R. J., PETERSON, E. L. & ZENER, C. (eds.) (1967). *Geometric Programming: Theory and Applications*. : Wiley, New York
- EBERHART R. & KENNEDY J. (1995). *A New Optimizer Using Particle Swarm Theory* Indiana, USA.
- ERSEL CANYURT O. (2005). Estimation of welded joint strength using genetic algorithm approach. *International Journal of Mechanical Sciences*, 47, pp. 1249-1261.
- FARNSWORTH M., BENKHELIFA E., TIWARI A., ZHU M. & MONIRI M. (2011). An efficient evolutionary multi-objective framework for MEMS design optimisation:

- validation, comparison and analysis. *Memetic Computing* 3, 175-197.
- FIACCO A. & MCCORMICK G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, New York.
- FONSECA C. M. & FLEMING P. J. (1993). Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion and Generalization Fifth International Conference on Genetic Algorithms. San Mateo, USA.
- FONSECA C. M. & FLEMING P. J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorithms. Part II: Application examples. *IEEE Transactions on systems, man and cybernetics*, 288, pp. 38-47.
- FRICKE W. (2003). Fatigue analysis of welded joints: state of development. *Marine and Structures*, 16, 185-200.
- GARWOOD S. J. (1997). Investigation of the Mv Kurdistan casualty Engineering Failure Analysis, 4, 3-24.
- GERE J. M. (2001). *Mechanics of Materials*, 5th ed., United States, Brooks/Cole.
- GOMORY R. E. (ed.) 1963. An all-integer integer programming algorithm, Chapter 13 in *Industrial Scheduling*: Prentice-Hall, Englewood Cliffs, NJ.
- GUNZBURER M. & BURKDART J. (2004). Uniform measures for point samples in hypercube [Online]. Available: <http://people.sc.fsu.edu/~burkardt/pdf/ptmeas.pdf> [Accessed 2 April 2011].
- GURBAL E. (2003). *Un weighted Multi-criteria Mesh and Structural Optimization Method with Finite Element Analysis*. Ph. D., Huddersfield.
- HOLLANDS J. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI, University of Michigan Press.
- HOWARD B C. (ed.) (1998). *Modern Welding Technology*.
- HUANG X. & MOAN T. (2007). Improved modeling of the effect of R-ratio on crack growth rate. *International Journal of Fatigue*, Vol. 29, 591-602.
- HURTIER P., JONES M. J., DESRAYAUD C., DRIVER, J. H., MONTHEILLET, F. & ALLEHAUX, D. (2006). Mechanical and thermal modelling of Friction Stir Welding *Journal of Materials Processing Technology*, 171, pp. 348-357.
- IBRAHIM K. (ed.) (2007). *Welding science and technology* new age international limited
- KIM D., RHEE S. & PARK H. (2002). Modeling and optimization of GMA welding process by genetic algorithm and response surface methodology. *International Journal of Production Research*. 40, 1699-1711.
- KIRKPATRICK S., GELATT C. & VECCHI M. (1983). *Optimization by Simulated Annealing. Science*, Vol. 220.
- KNOWLES J. & CORNE D. (1998). The Pareto archived evolution strategy: A new baseline algorithm for Pareto multi-objective optimization. *Parallel Problem Solving from Nature. PPSN V. 5th International Conference*.

- KRISTOFFER K. (2012). *Probabilistic modeling of fatigue failures*. Doctoral thesis no. 79, 2012 KTH School of Engineering Sciences.
- LANCASTER F. (1987). *Metallurgy of Welding*.
- LOCKE E. V., HOAG, E. D. & HELLA, R. A. (1972). Deep penetration welding with high power CO₂ lasers. *IEEE Journal of Quantum Electronics*, QE-8, 132–5.
- LOCKHART S. D. & JOHNSON, C. (eds.) (1996). *Engineering design communication*. : Reading, MA: Addison-Wesley.
- LOTSBERG I. (2004). Fatigue design of welded pipe penetrations in plated structures *Journal of Marine Structures* 17, 29-51.
- MODEFRONTIER 2008. modeFRONTIER v4.0 User manual. Trierste, Italy: ESTECO srl.
- MOLENT L., JONES R., BARTER S. & PITT S. (2006). Recent development in fatigue crack growth assessment. *International Journal of Fatigue*, Vol. 28, 1759-1768.
- MORIS A. (ed.) (1982). *Foundations of Structural Optimization*, New York: Wiley.
- NASIR A. (ed.) (2005). *New development in advanced welding*, Abington, Cambridge: Woodhead Publishing Limited
- NEWMAN J. C. & EDWARDS, P. R. (1988). Short-Crack Growth Behavior in an Aluminum Alloy - and AGARD Cooperative Test Programme AGARD R-732. AGARD R-732.
- NEWMAN J. C., WU X. R., VENNARI S. & LI, C. (1994). Small-Crack Effects in High-Strength Aluminum Alloys. NASA RP-1309.
- NGUYEN T. & WAHAB M. A. (1998). The effect of weld geometry and residual stresses on the fatigue of welded joints under combined loading. *Journal of Materials Processing Technology*, 77, 201-208.
- PARIS P. C. & ERDOGAN, F. (1960). A Critical Analysis of Crack Propagation Laws. *Journal of Basic Engineering*, 85, 528-534.
- PIGNEAUX D. (2002). *Effet d'entaille en fatigue dans les joints soudés en aciers*. (In French). . PhD. Thesis, University de Metz.
- POLES S. (2003). MOGA-II An improved Multi-Objective Genetic Algorithm. Trieste, Italy. ESTECO Technical Report.
- POLES S., RIGONI E. & ROBIC T. (2004). MOGA-II Performance on Noisy Optimization Problems International conference on Bio inspired Optimization
- POLI R., LANGDON W. & MCPHEE M. (2008). *A field guide to genetic programming*. [Online]. Available: <http://www.gp-field-guide.org.uk> [Accessed 15 Feb 2010].
- POLONI C. & PEDIRODA, V. (1997). GA coupled with computationally expensive simulations: tools to improve efficiency. *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*. John Wiley and Sons, England.
- PYTTEL B., GRAWENHOF P. & BERGER C. (2012). Application of different concepts for fatigue design of welded joints in rotating components in mechanical engineering

International Journal of Fatigue, 34, 35-46.

- RADAJ D. (1996). Review of fatigue strength assessment of no welded and welded structures based on local parameters *International Journal of fatigue*, 18, 153-170.
- RADAJ D., SONSINO C. & FRICKE W. (2006). *Fatigue assessment of welded joints by local approaches*, Cambridge: Woodhead Publishing Ltd.
- RADAJ D., SONSINO C. & FRICKE W. (2009). Recent developments in local concepts of fatigue assessment of welded joints. *International Journal of Fatigue*, 31, 2-11.
- RANUT, D. P. (2012). Optimization and inverse problems in heat transfer Dottorato di Ricerca in Technologie Chimicheed Energetiche, Universita Degli Studi di Udine, Trieste-Italia.
- RAO S. (ed.) (2009). *Engineering Optimization*: Wiley-Interscience Publications.
- REEDY J. N. (ed.) (1993). *An Introduction to the Finite Element Method*: Mc Graw-Hill, Inc
- RIGONI E. (2010). Fast optimizers: General description. Technical Report 2010-004. ESTECO s.r.l. Italy.
- RIGONI E. & POLES S. (2005). NBI and MOGA-II two complementary algorithms for Multi-Objective optimizations, Practical Approach to Multi-Objective Optimization. IBFI, Schol Dagstuhl, Germany.
- RIGONI E. & TURCO, A. (2010). Metamodels for fast multi-objective optimization: trading off global exploration and local exploitation Simulated Evolution and Learning Lecture Notes in Computer Science, 6457, 523-532.
- SAHIN M. & EROL AKATA H. (2003). Joining with friction welding of plastically deformed steel. *Journal of Materials Processing Technology*, 142, 239-246.
- SANAZ M. (2004). *Multi-Objective Evolutionary Algorithm: Data structures, Convergence and Diversity* Ph. D., Shaker Verlag.
- SBALZARINI I. F., MULLER S. & KOUMOUTSAKOS P. (2009). Multi-objective optimization using evolutionary algorithms. [Online]. Available: <http://www.mosaic.ethz.ch/research/docs/Sbalzarini2000b.pdf> [Accessed 22 March 2012].
- SCHAFFER J. (1987). Multi-objective Optimization with Vector Evaluated Genetic Algorithms. In: *Proceeding of the 1st International Conference on Genetic Algorithms*, U.S.A.
- SCHIJVE J. (1979). Four Lectures on Fatigue Crack Growth. *Engineering Fracture Mechanics*, 11, 167-221.
- SCHILLER S. (ed.) 1977. *Elektronenstrahltechnologie*, Stuttgart: Wissenschaftliche Verlagsgesellschaft.
- SCOTTI A. & ALBUQUERQUE L. A. (1997). Influence of oscillation parameters on crack formation in automatic Fe-B hard facing. *Journal of Materials Processing Technology*, 65, pp. 272-280.
- SECHADRI A. (2006). *Statistical variation of weld profiles and their influence on fatigue strength*. MSc. Thesis, Lappeenranta University of Technology.
- SEDENKA V. & RAIDA Z. (2010). Critical Comparison of Multi-objective Optimization

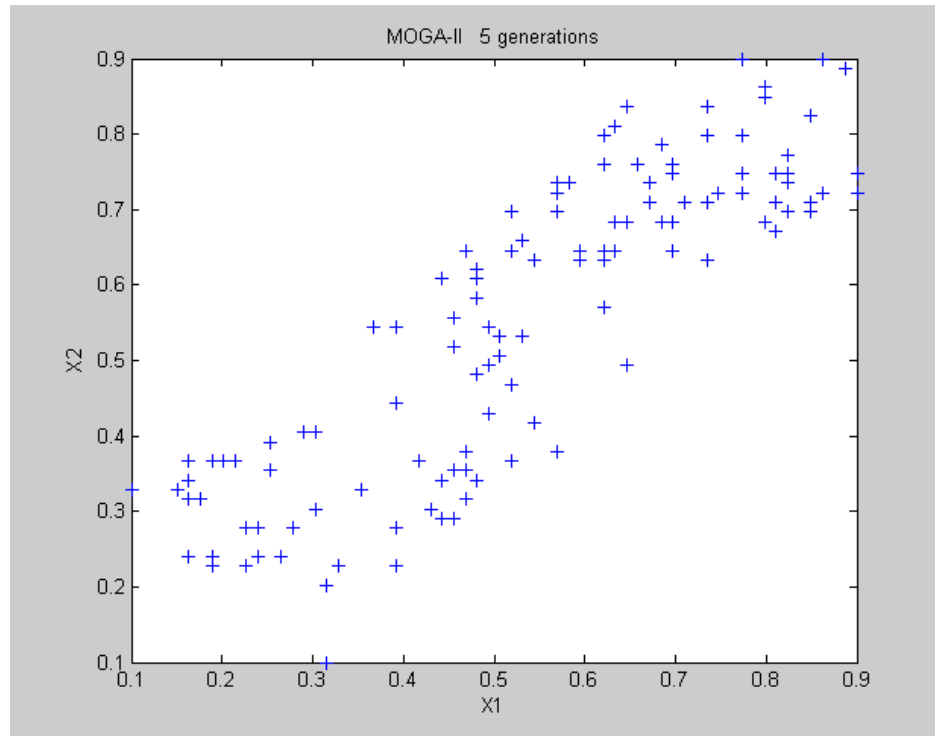
- Methods: Genetic Algorithms versus Swarm Intelligence. *Radio engineering*, 19, pp. 369-377.
- SHINGAWA K. & KU S. (2001). *Weld imperfections*, Japan, KOBE Steel Ltd.
- SHUKLA P. K., DEB K. & TIWARI. (2004). *Comparing classical generating methods with evolutionary multi-objective optimization methods*. [Online]. Available: <http://www.iitk.ac.in/kangal/papers/k2004015.pdf> [Accessed 5 April 2012].
- STATNIKOV R. & MATUSOV J. (2012). Multi-criteria Engineering Optimization Problems: Statement Solution and Applications *Journal of Optimization Theory and Applications*.
- STEPHENS R. I., FATEMI A., STEPHENS R. R. & FUCHS H. O. (2001). *Metal Fatigue in Engineering*, New York, John Wiley & Sons.
- STERJOVSKI Z., NOLAN D., CARPENTER K. R., DUNNE D. P. & NORRISH J. (2005). Artificial neural networks for modelling the mechanical properties of steels in various applications. *Journal of Materials Processing Technology*, 170, pp. 536-544.
- STROMBERG N. (2010). Nonlinear FEA and Design Optimization for Mechanical Engineering. Jonkoping, Jonkoping University, School of Engineering.
- SUDHAKARAN R., MURUGAN V. V., KUMAR K. M., JAYARAM R., PUSHPARAJ A., PRAVEEN C. & VENKATI V. N. (2011). Effect of welding process parameters on weld bead geometry and optimization of process parameters to maximize depth to width ratio for stainless steel gas tungsten arc welded plates using genetic algorithm. . *European Journal of Scientific Research*, 62, 76-94.
- SUPPAPINARM A., SEFFEN K. A. & PARKS G. T. (2000). A simulated Annealing Algorithm for a Multi-objective Optimization. *Engineering Optimization*, 33, pp. 59-85.
- TAHERNEZHADIANI K., HAMZEH A. & HASHEMI S. (2012). Toward Enhancing Solution Space Diversity in Multi-objective Optimization: A hyper volume-based approach. . *International Journal of Artificial Intelligence and Applications*, 3.
- TAGUSHI G. (1993). *On Robust Technology Development: Bringing Quality Engineering Upstream*. New York: ASME Press.
- TENG T., FUNG C. & CHANG P. (2002). Effect of weld geometry and residual stresses on fatigue in butt-welded joints, *International Journal of Pressure Vessels and Piping*, 79, 467-482.
- ULTMAN D. G. (1994). *The Mechanical Design process*, McGraw Hill.
- VASUDEVAN M., BHADURI A. K., RAJ B. & PRASAD K. (2007). Genetic algorithm based computational models for optimizing the process parameters of A-TIG welding to achieve target bead geometry in type 304L (N) and 316 L (N) stainless steels. *Materials and Manufacturing Processes*, 22, 641-649.
- WEI L. & PENTTI M. (2003). *Advanced Steel Structures*. Espo Finland: Helsinki University of Technology.
- WELDING A. C. O. L. B. (1983). *Material Hand Book*.
- WILLIAM D. & CALLISTER J. (2003). *Material science and engineering*. 6th edition, John Wiley.

- YOON H. K., KONG Y. S., KIM S. J. & KOHYAMA A. (2006). Mechanical properties of friction welds of RAFs (JLF-1) to SUS304 steels measured by the acoustic emission technique. *Journal of Fusion Engineering and Design*, 81, pp. 945-950.
- ZITZLER E. (1999). Evolutionary algorithms for multi-objective optimization: methods and applications. [Online]. Available: <http://www.tik.ee.ethz.ch/sop/publicationListFiles/zitzl1999a.pdf> [Accessed 5 April 2012].
- ZITZLER E., DEB K. & THIELE L. (2000). Comparison of multi-objective evolutionary algorithms: Empirical results. *Journal of Evolutionary Computation*, 8, 173-195.
- ZITZLER E. & THIELE L. (1998). An evolutionary algorithm for multi-objective optimization: the Strength Pareto Approach. . Zurich. Swiss Federal Institute of Technology.
- ZITZLER E. & THIELE L. (1999). Multi-objective evolutionary algorithms: a comparative case study and Strength Pareto approach. . *IEEE Transaction on Evolutionary Computation*, 3, 257-271.

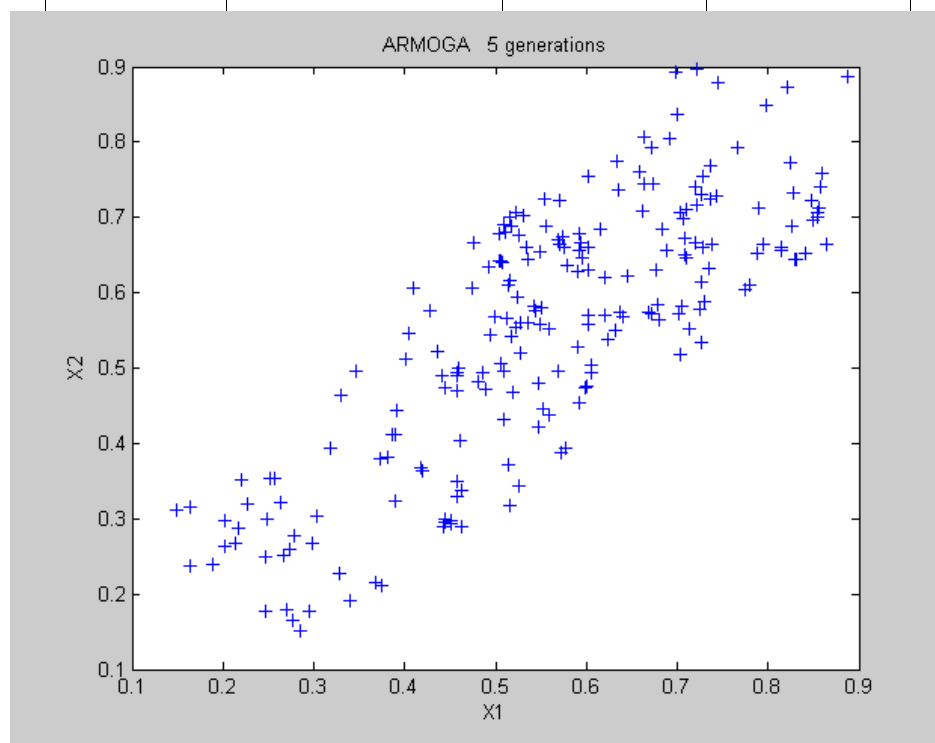
APPENDICES

APPENDIX A

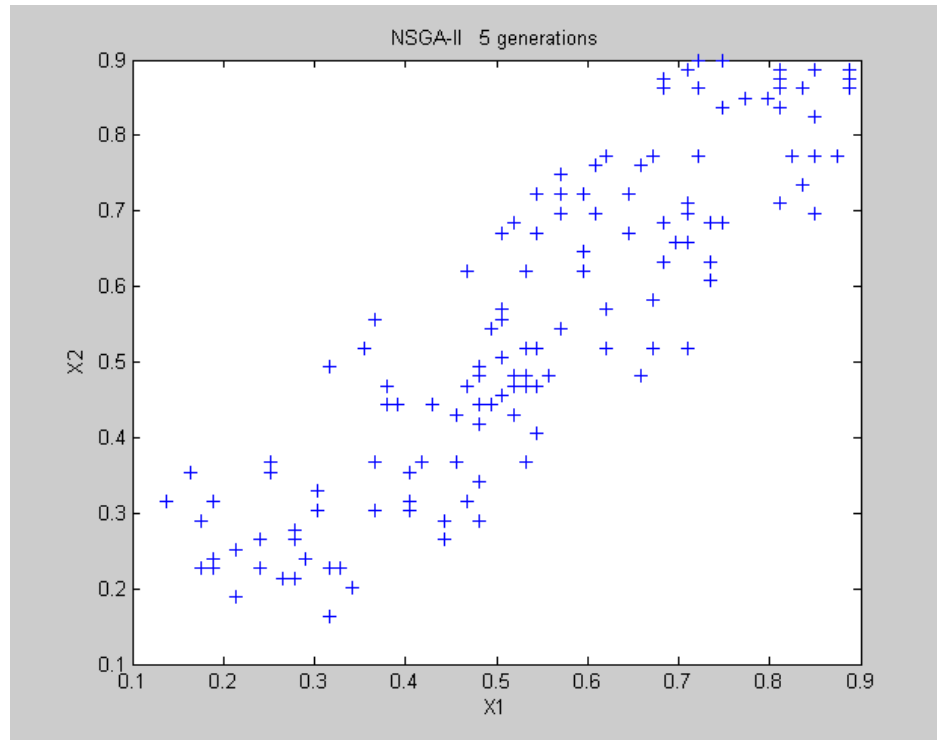
Graphical representation of Pareto set in design space with Pair Wise(PW) and Circumscription (CM) metrics for lifting arm problem with 5, 10, 15, 20, 25 and 30 generations



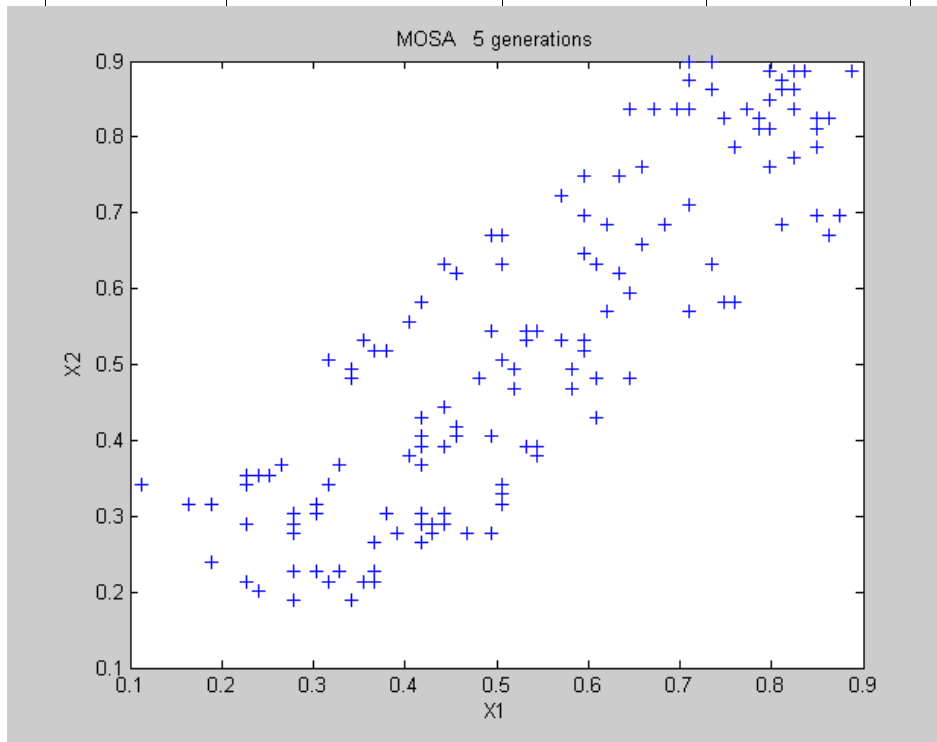
Mean	Standard deviation	PW-metric	CM-metric
0.025686	0.015665	0.6098653	1.6136



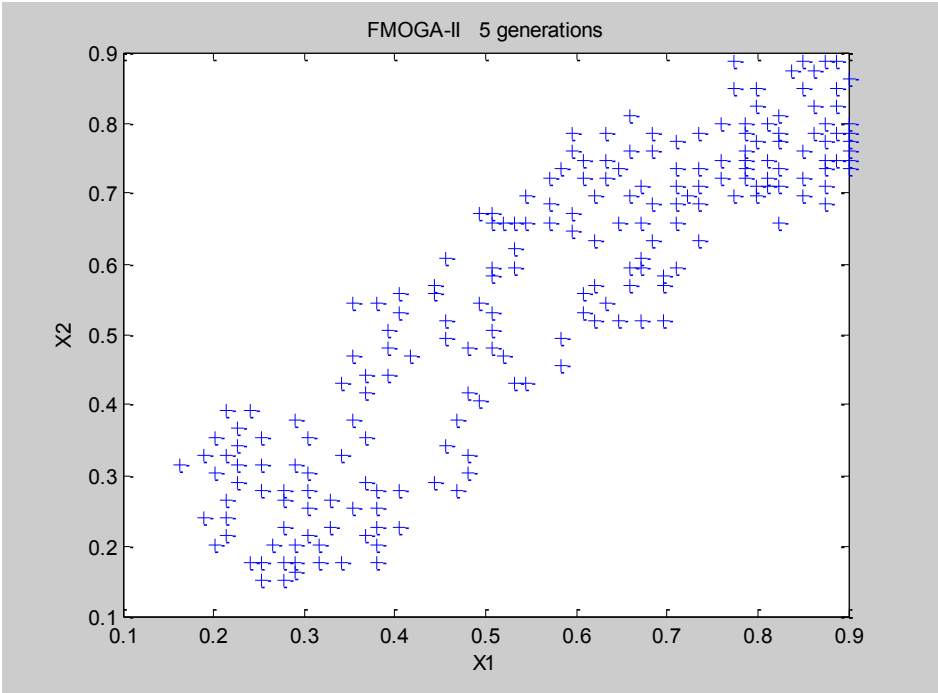
Mean	Standard deviation	PW-metric	CM-metric
0.016397	0.011432	0.697201	1.6122



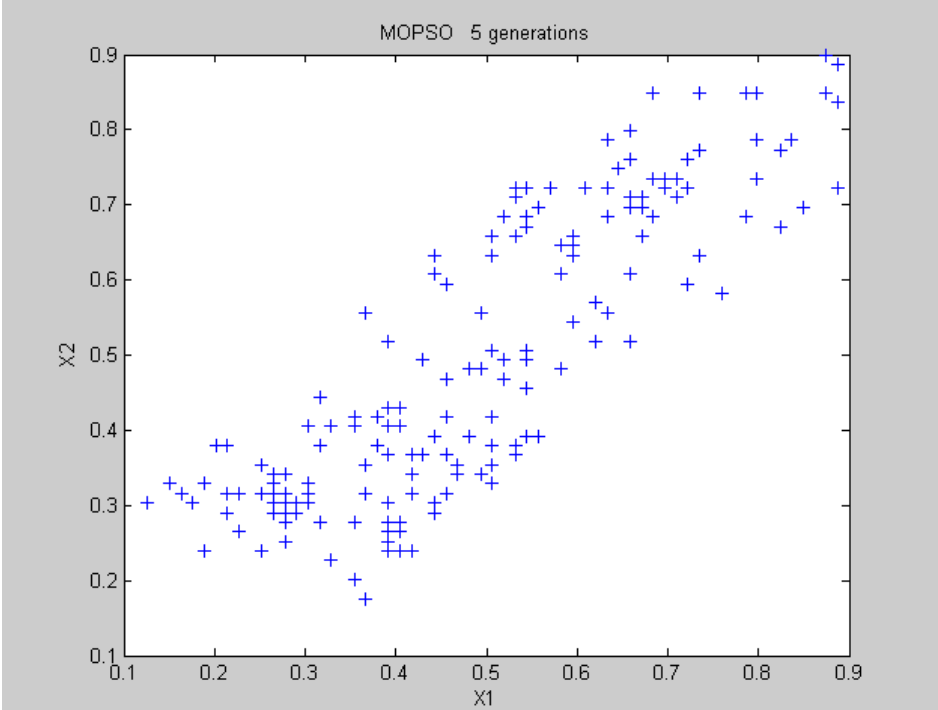
Mean	Standard deviation	PW-metric	CM-metric
0.02466	0.01176	0.476886	1.6125



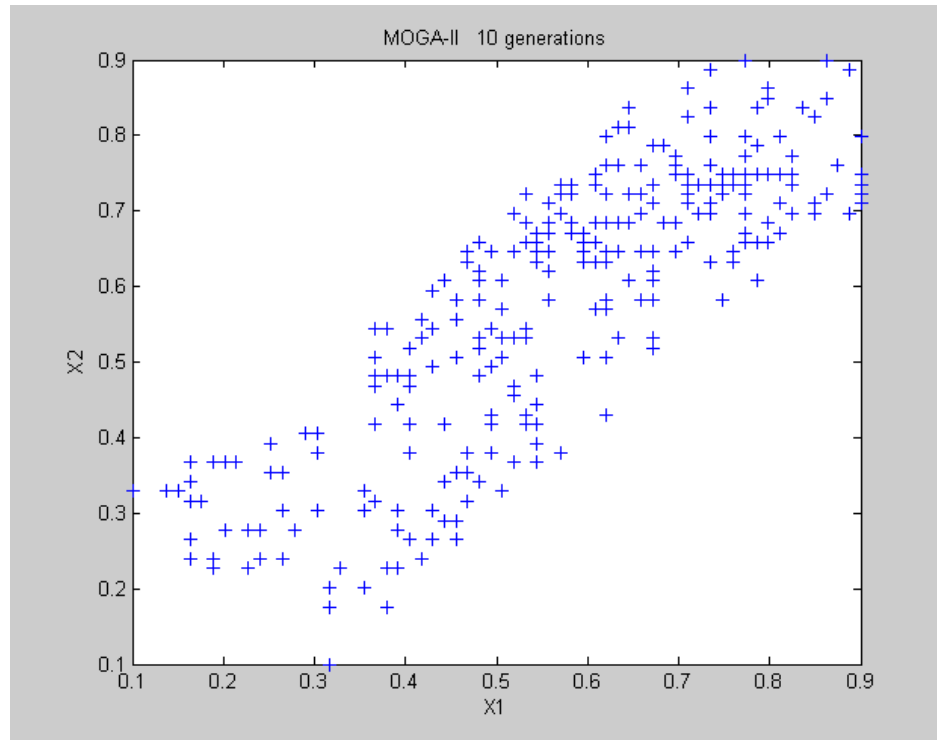
Mean	Standard deviation	PW-metric	CM-metric
0.022123	0.010246	0.463138	1.5977



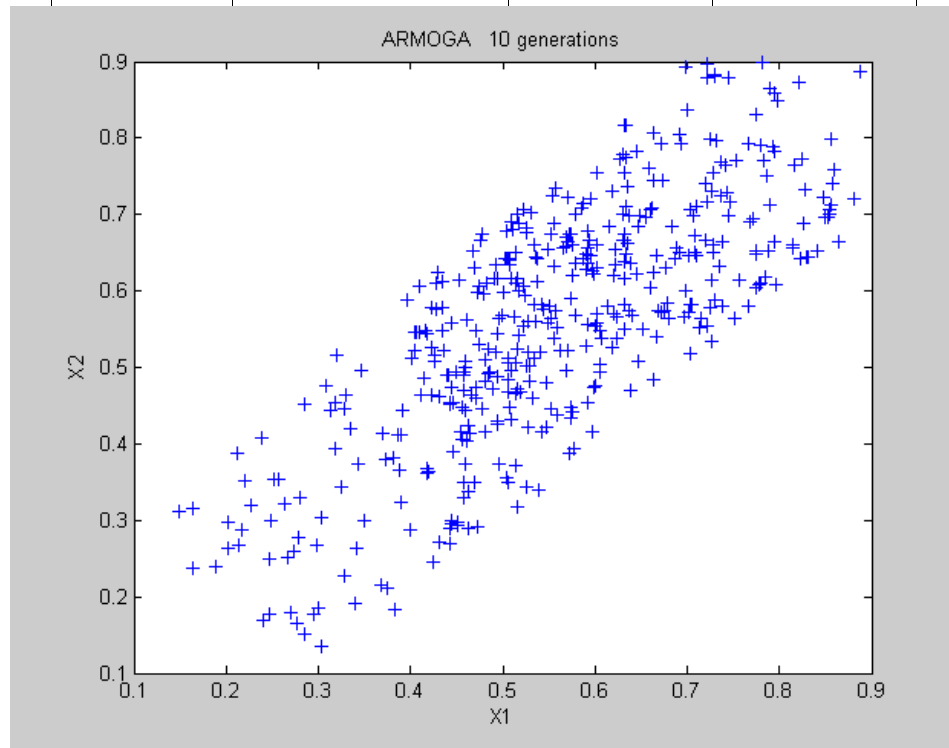
Mean	Standard deviation	PW-metric	CM-metric
0.021486	0.006764	0.31481	1.6389



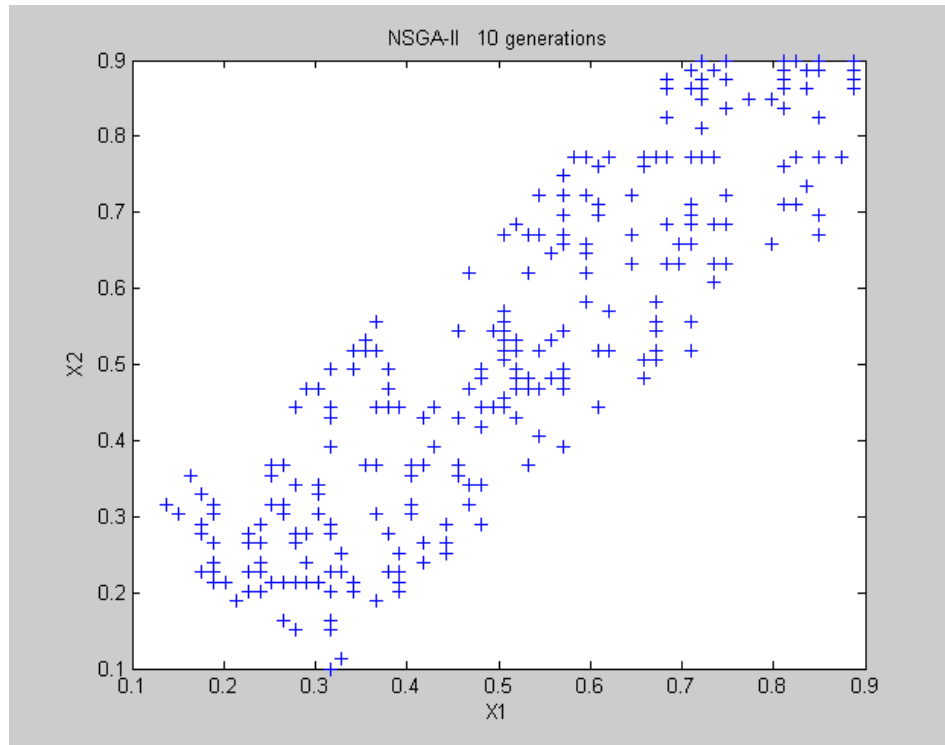
Mean	Standard deviation	PW-metric	CM-metric
0.021486	0.010478	0.487666	1.6265



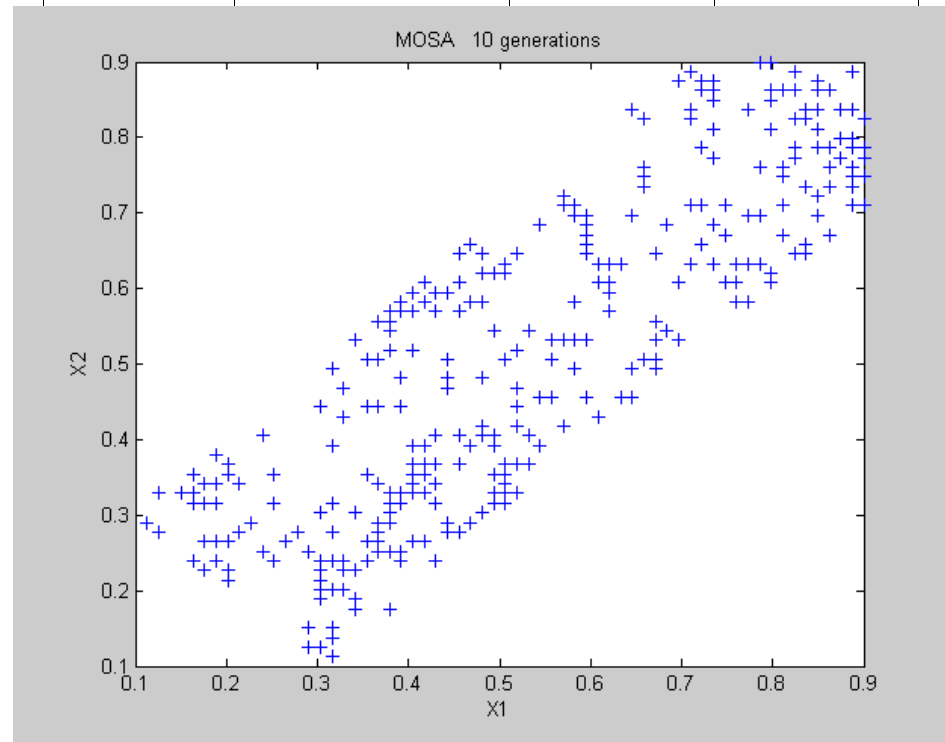
Mean	Standard deviation	PW-metric	CM-metric
0.018635	0.008997	0.482801	1.6503



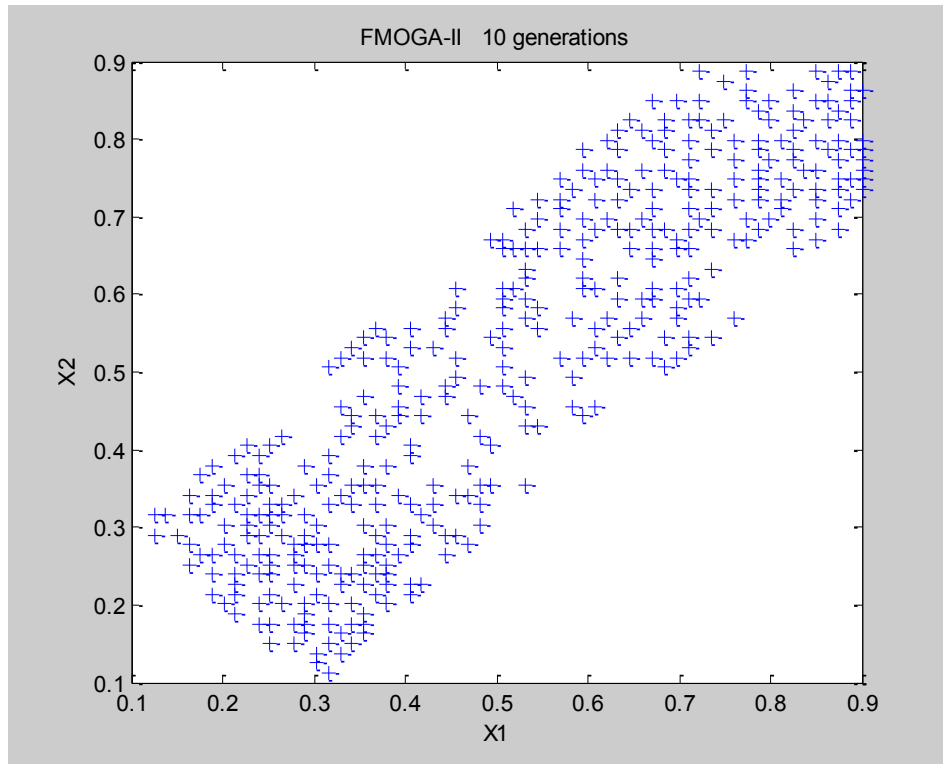
Mean	Standard deviation	PW-metric	CM-metric
0.01184	0.008524	0.719932	1.6378



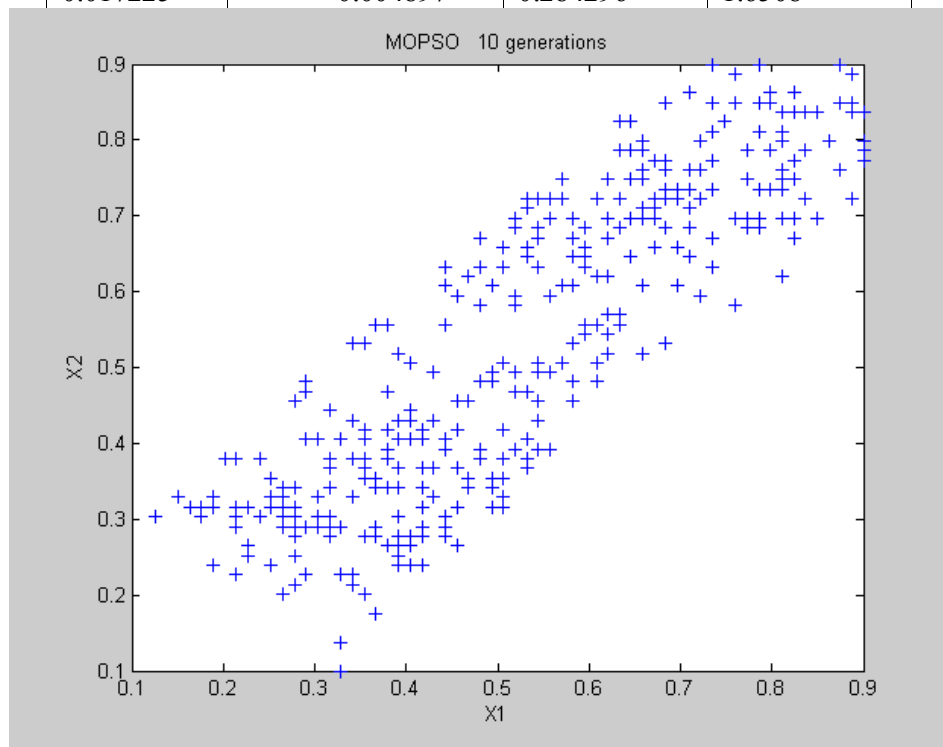
Mean	Standard deviation	PW-metric	CM-metric
0.01777	0.008309	0.467586	1.6515



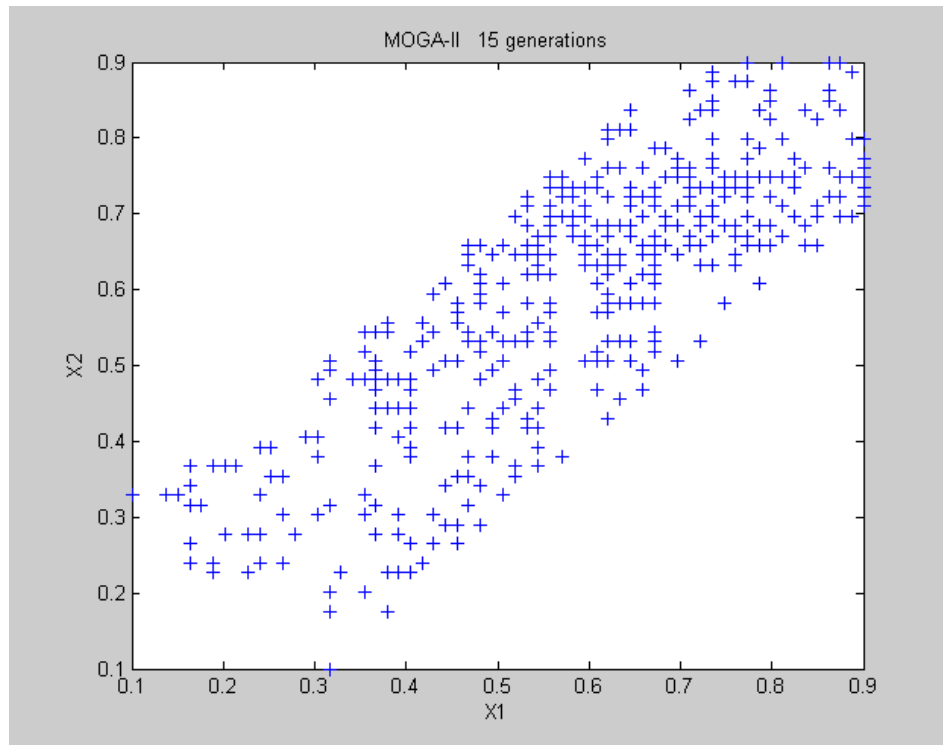
Mean	Standard deviation	PW-metric	CM-metric
0.016897	0.006911	0.409008	1.6386



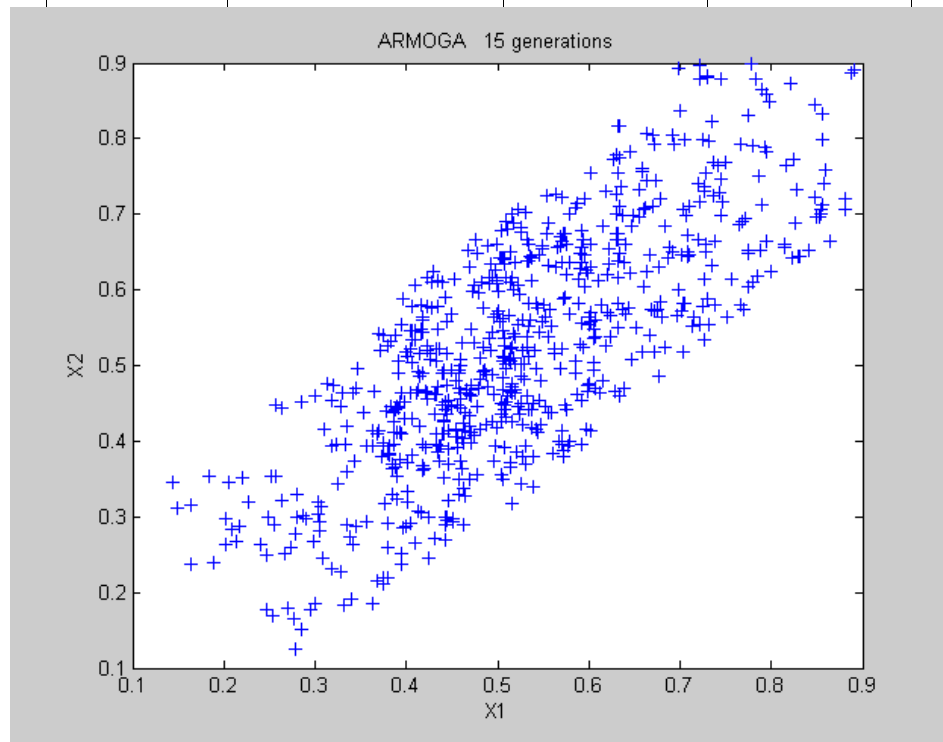
Mean	Standard deviation	PW-metric	CM-metric
0.017225	0.004897	0.284296	1.6508



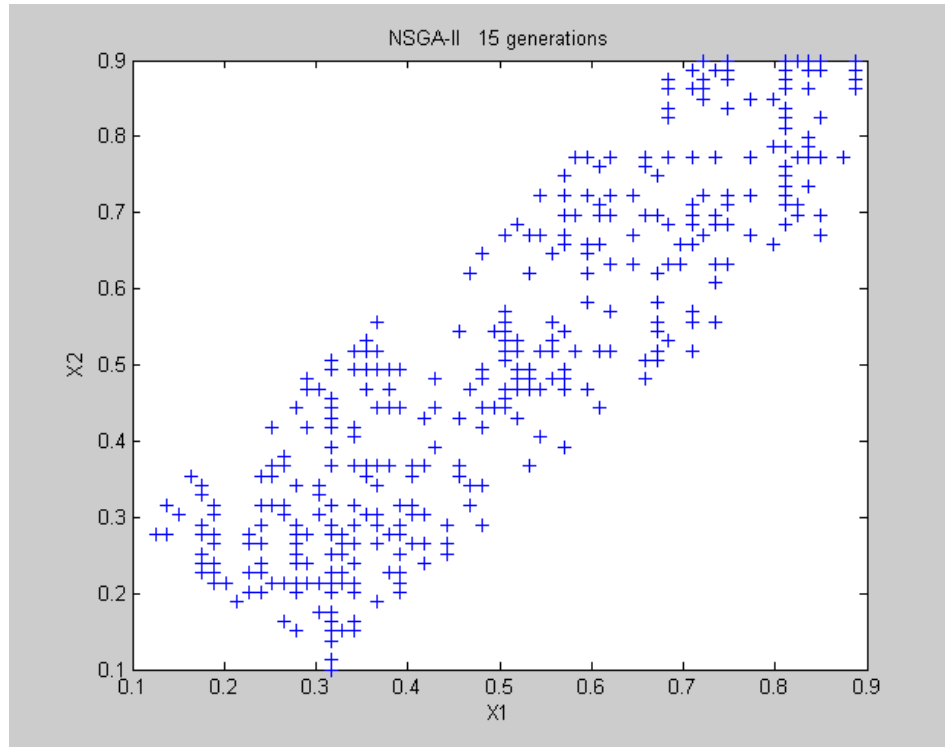
Mean	Standard deviation	PW-metric	CM-metric
0.01684	0.006592	0.391449	1.6506



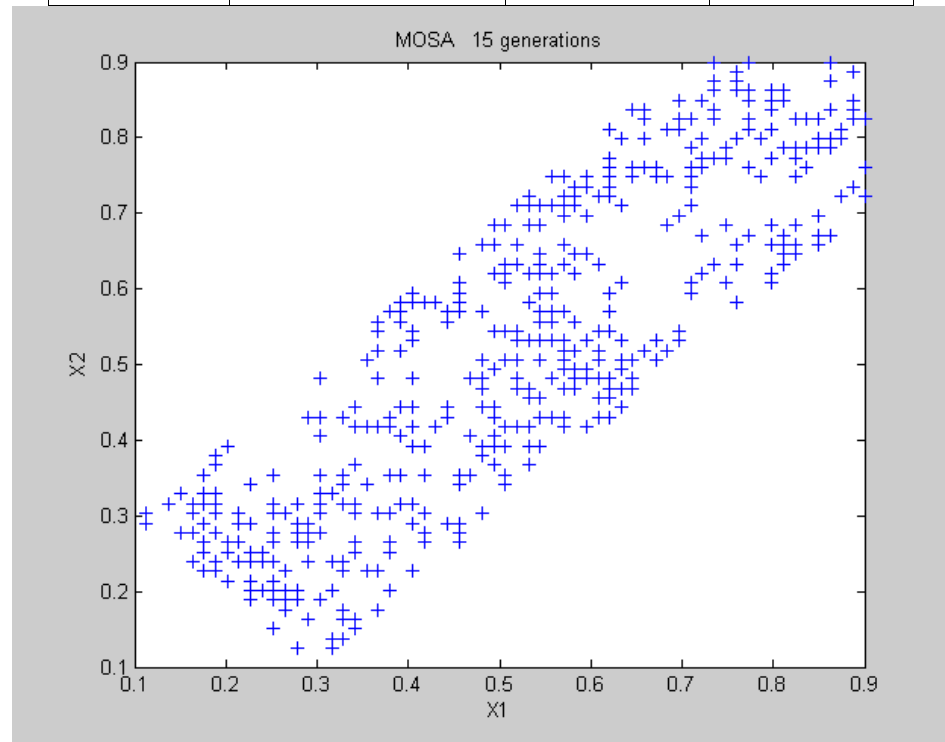
Mean	Standard deviation	PW-metric	CM-metric
0.015922	0.006453	0.405288	1.6609



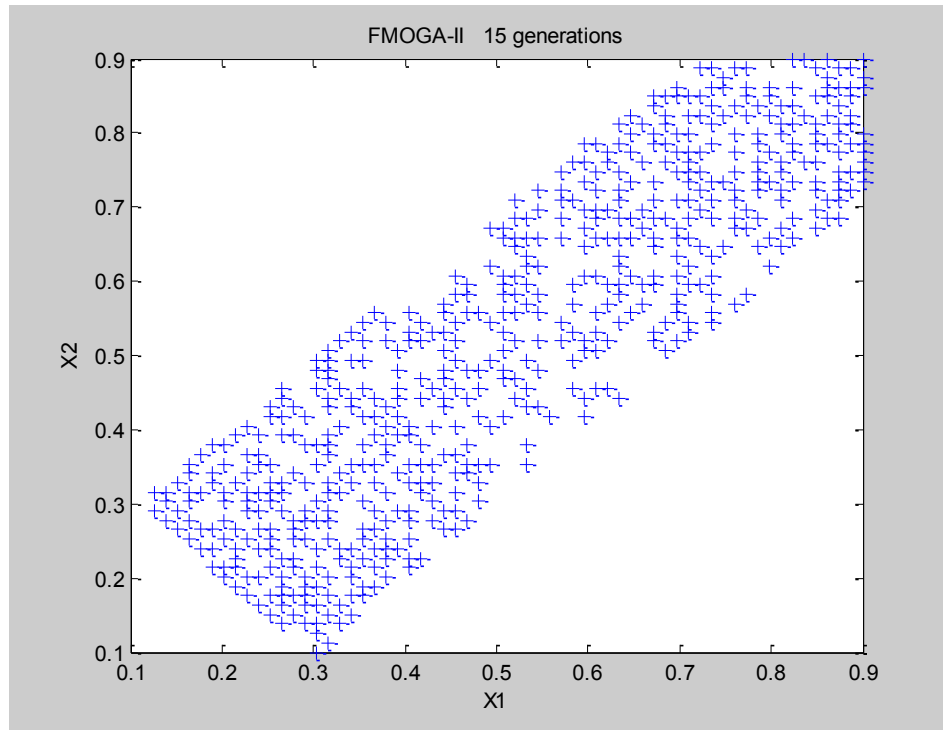
Mean	Standard deviation	PW-metric	CM-metric
0.008763	0.005815	0.663586	1.6491



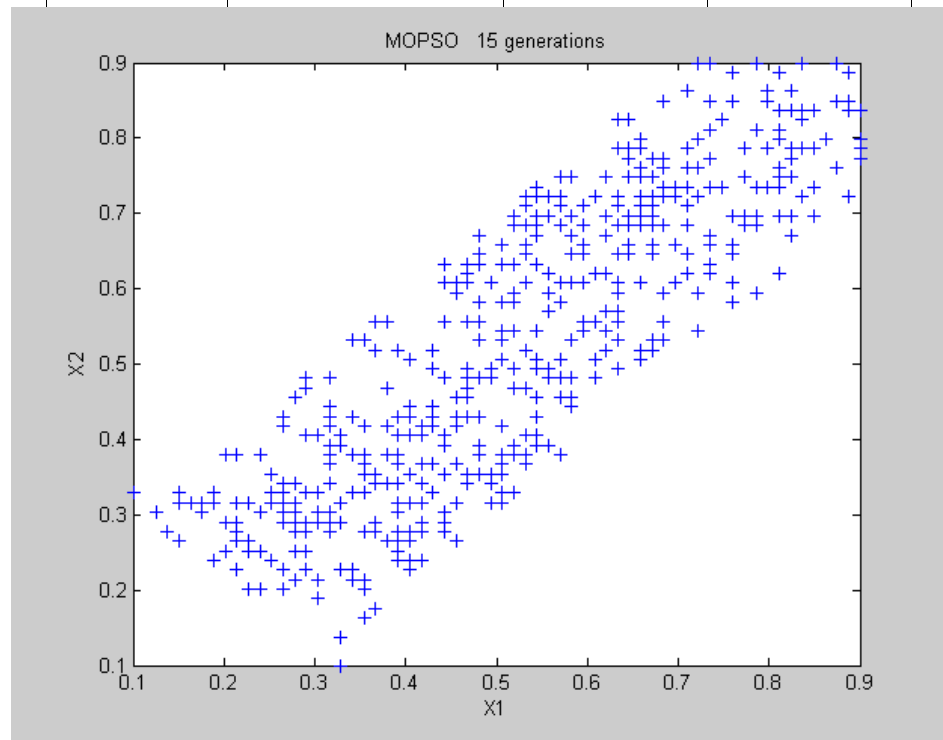
Mean	Standard deviation	PW-metric	CM-metric
0.015862	0.005755	0.362817	1.662



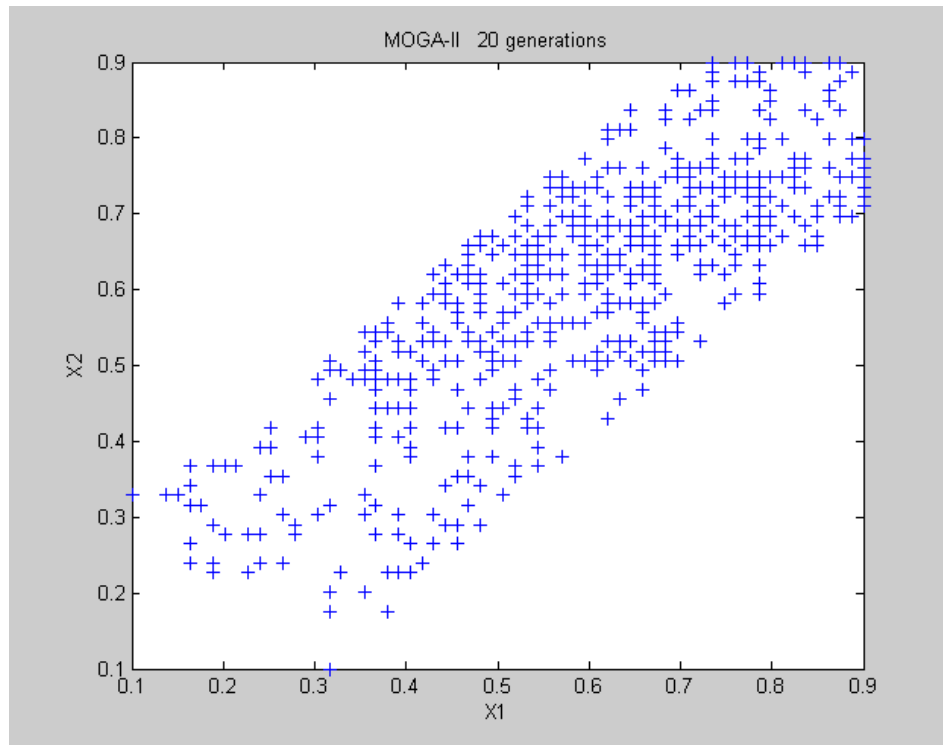
Mean	Standard deviation	PW-metric	CM-metric
0.01499	0.004788	0.319413	1.6499



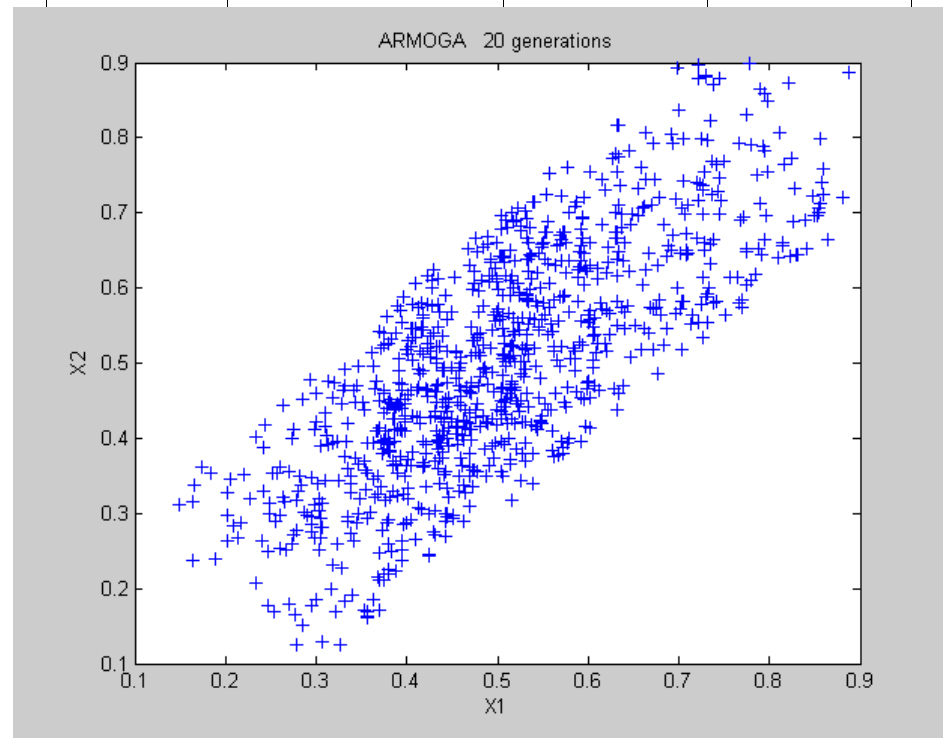
Mean	Standard deviation	PW-metric	CM-metric
0.014945	0.003187	0.213249	1.6622



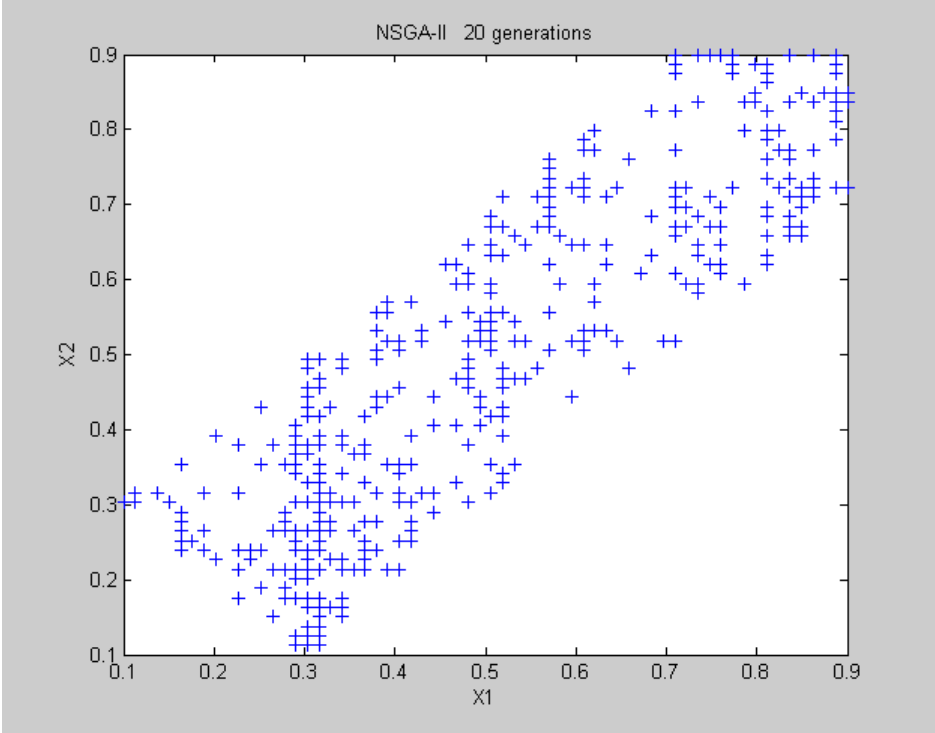
Mean	Standard deviation	PW-metric	CM-metric
0.015349	0.0051	0.332269	1.6618



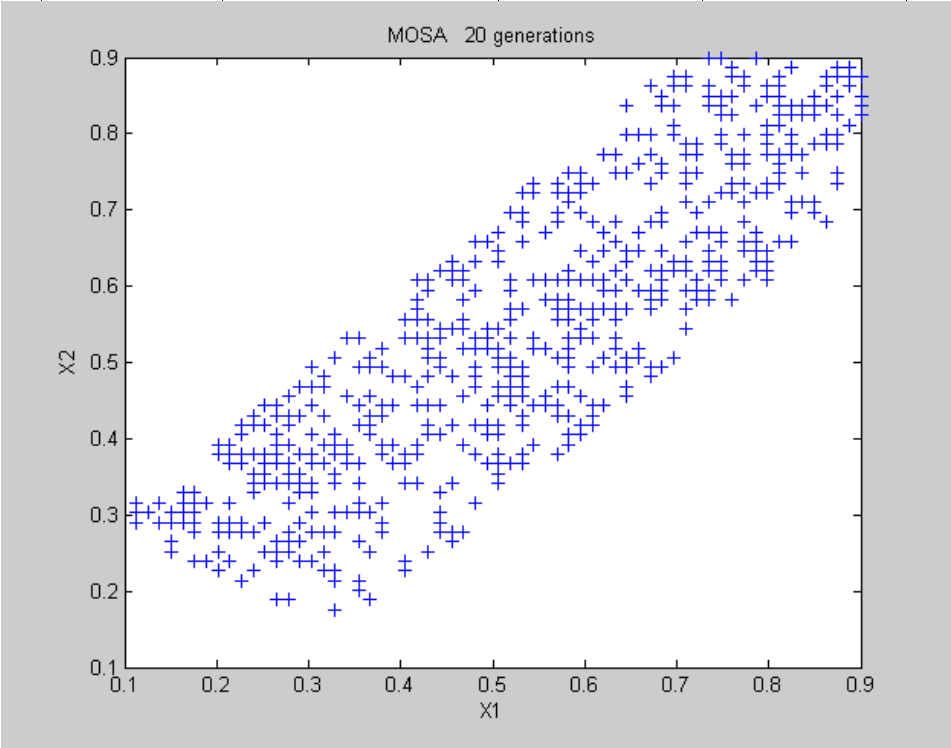
Mean	Standard deviation	PW-metric	CM-metric
0.014811	0.005228	0.352981	1.6506



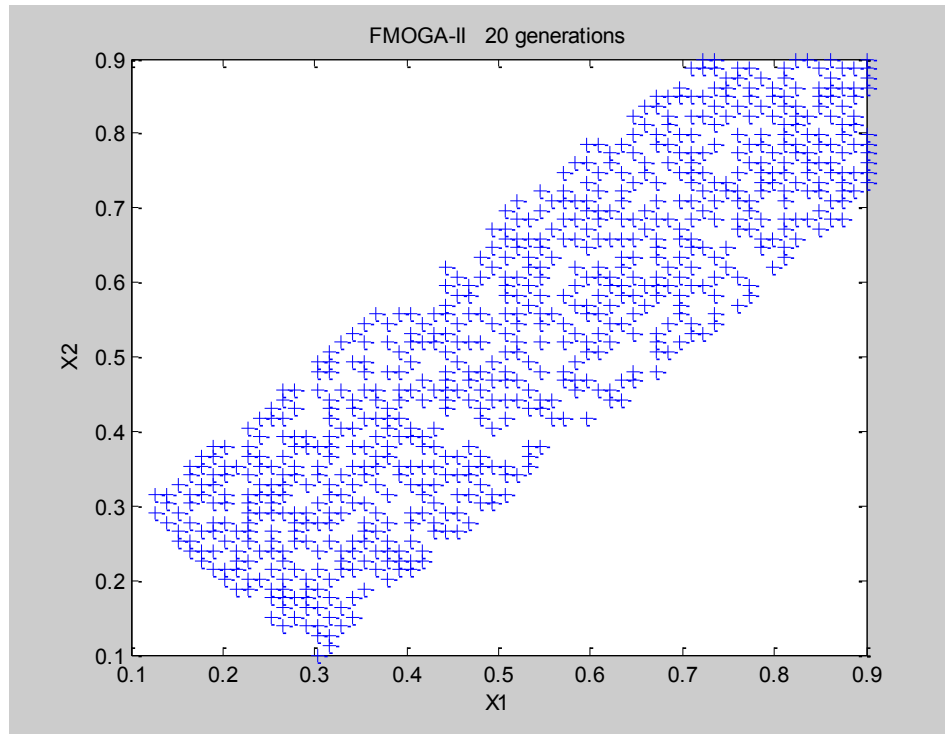
Mean	Standard deviation	PW-metric	CM-metric
0.007923	0.005626	0.710085	1.6491



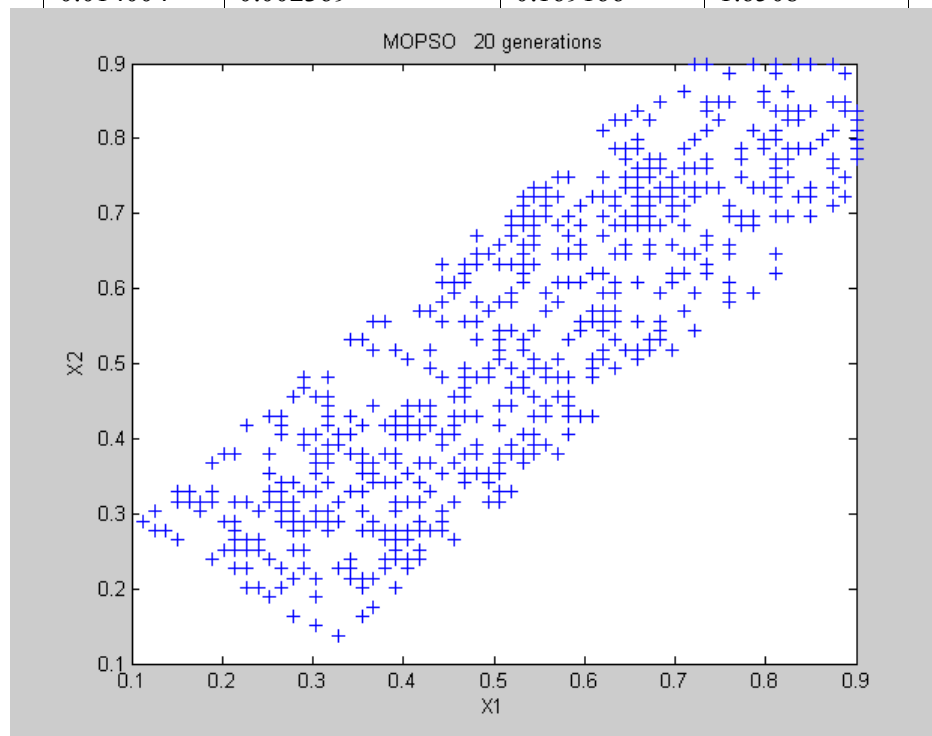
Mean	Standard deviation	PW-metric	CM-metric
0.015713	0.006528	0.415452	1.6505



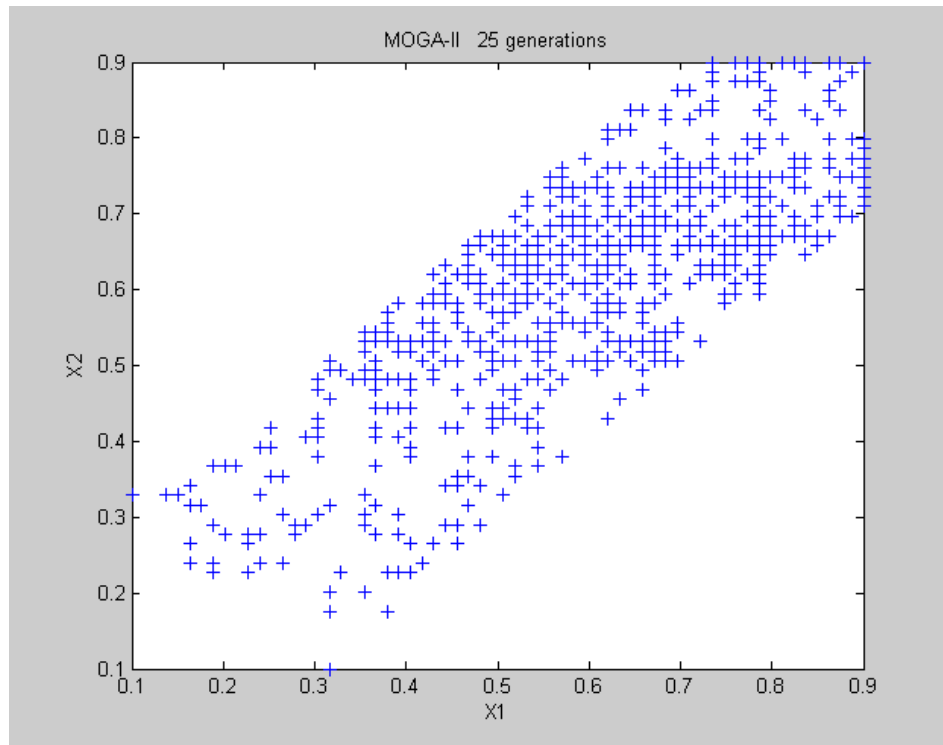
Mean	Standard deviation	PW-metric	CM-metric
0.013812	0.003006	0.217637	1.6383



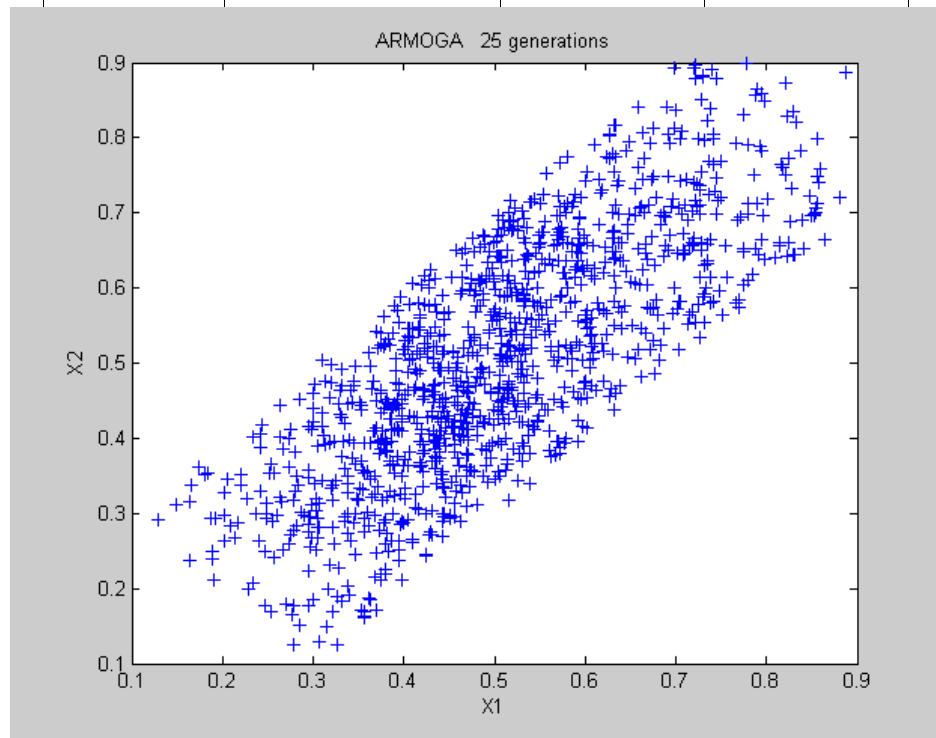
Mean	Standard deviation	PW-metric	CM-metric
0.014004	0.002369	0.169166	1.6508



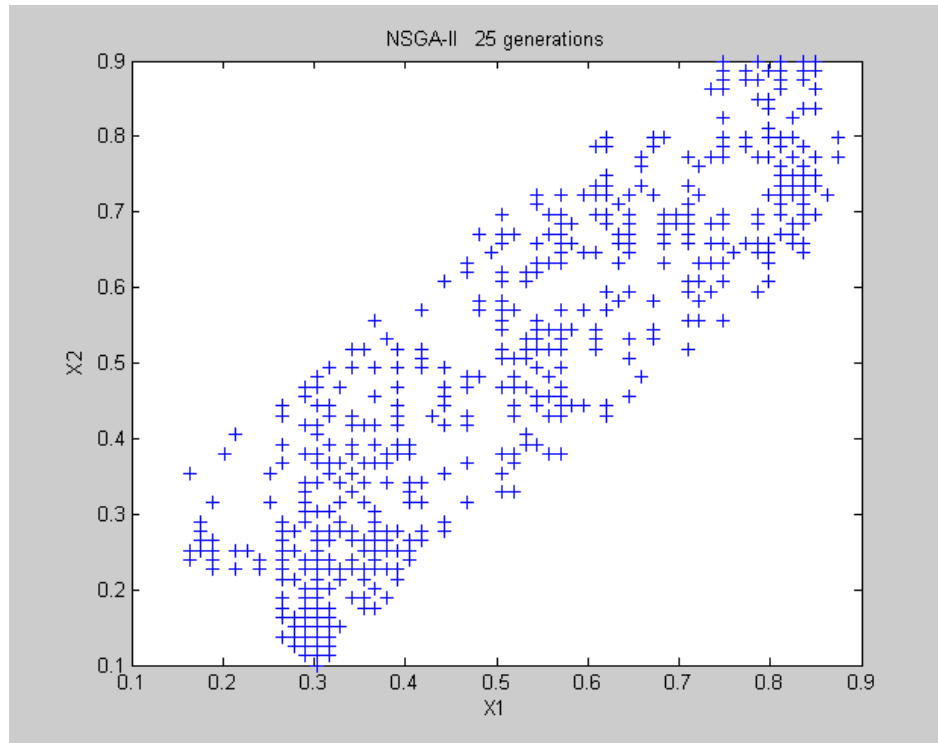
Mean	Standard deviation	PW-metric	CM-metric
0.014291	0.003591	0.251277	1.6496



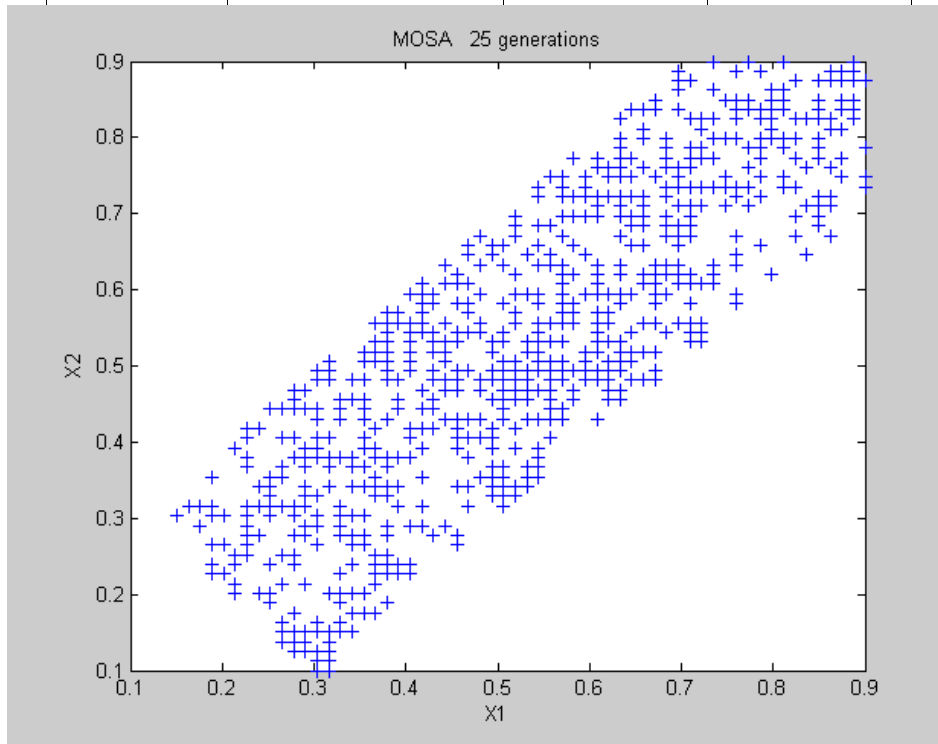
Mean	Standard deviation	PW-metric	CM-metric
0.014248	0.004739	0.332608	1.6513



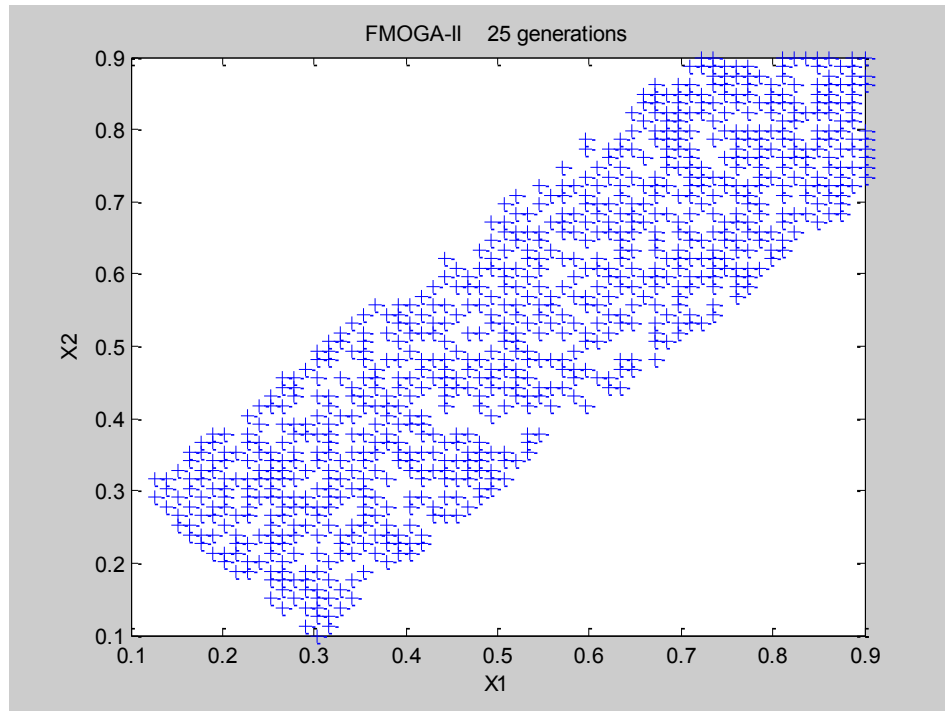
Mean	Standard deviation	PW-metric	CM-metric
0.00713	0.004987	0.698557	1.6491



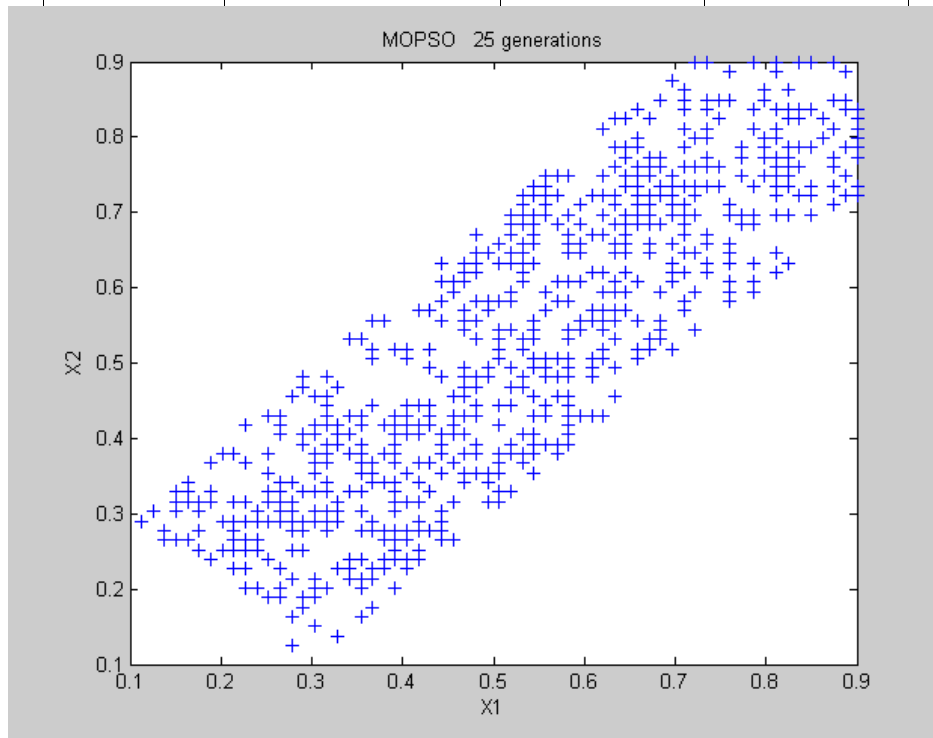
Mean	Standard deviation	PW-metric	CM-metric
0.014527	0.004594	0.316239	1.6499



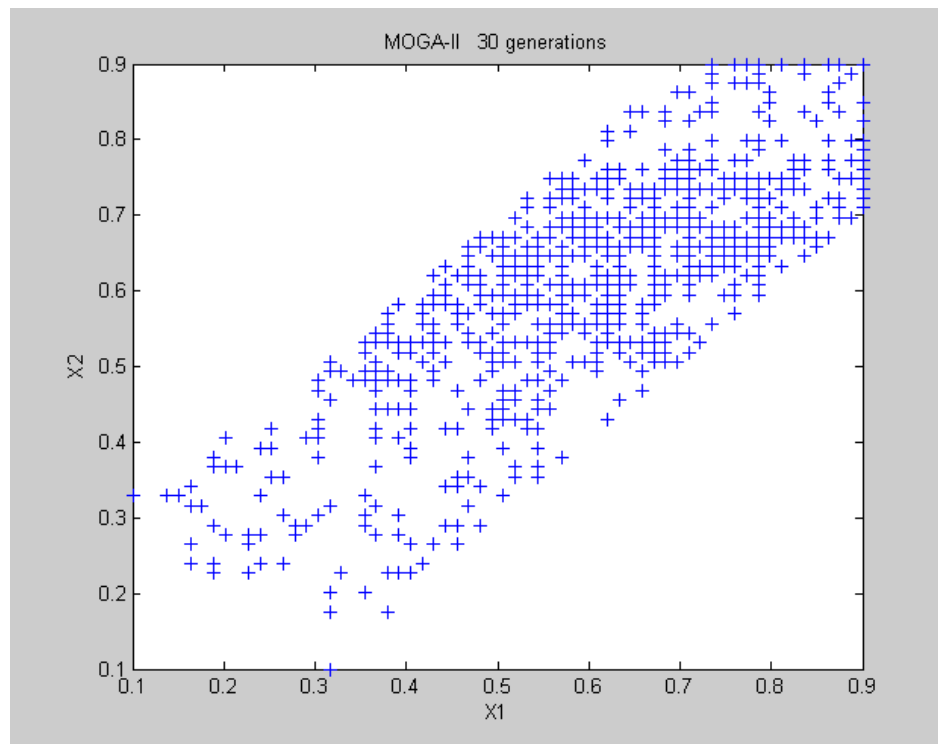
Mean	Standard deviation	PW-metric	CM-metric
0.013576	0.002914	0.214643	1.6497



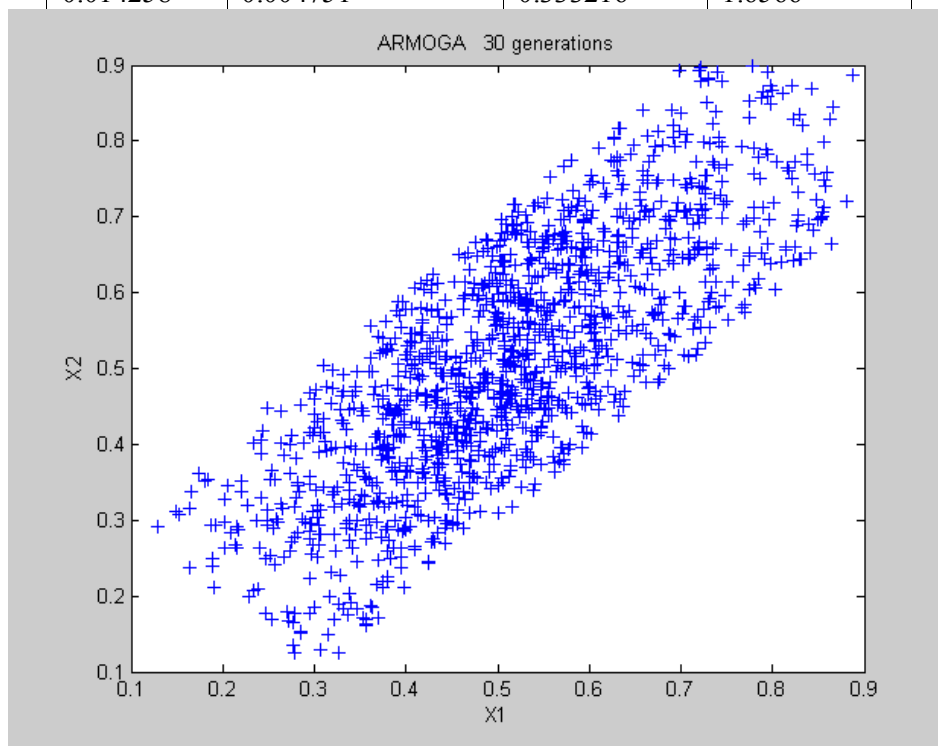
Mean	Standard deviation	PW-metric	CM-metric
0.013511	0.001967	0.145585	1.6514



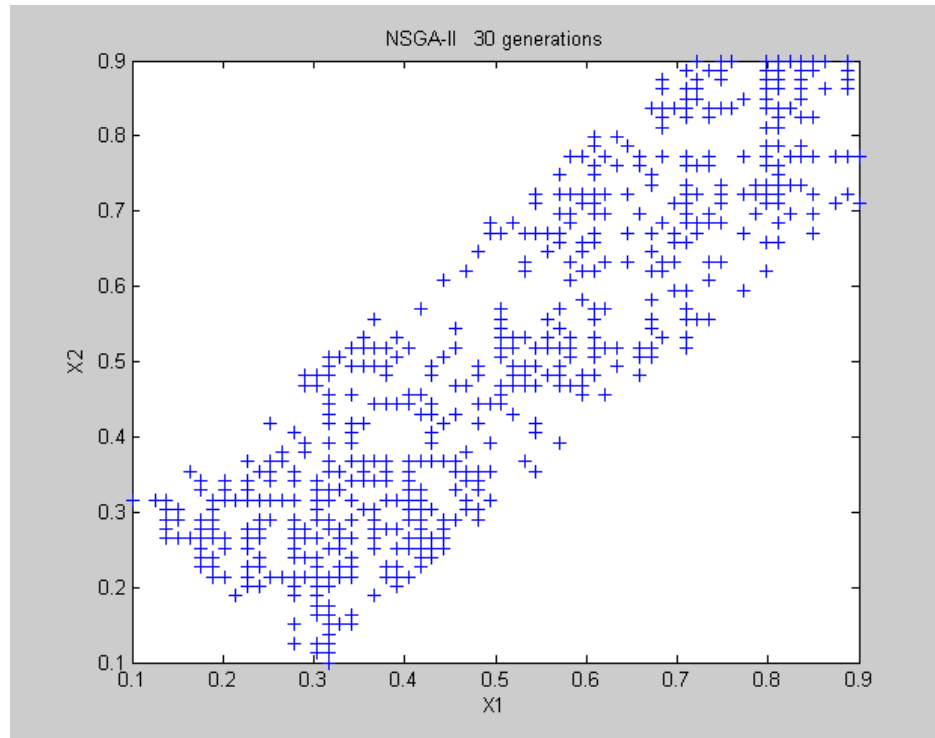
Mean	Standard deviation	PW-metric	CM-metric
0.013782	0.002951	0.21412	1.6386



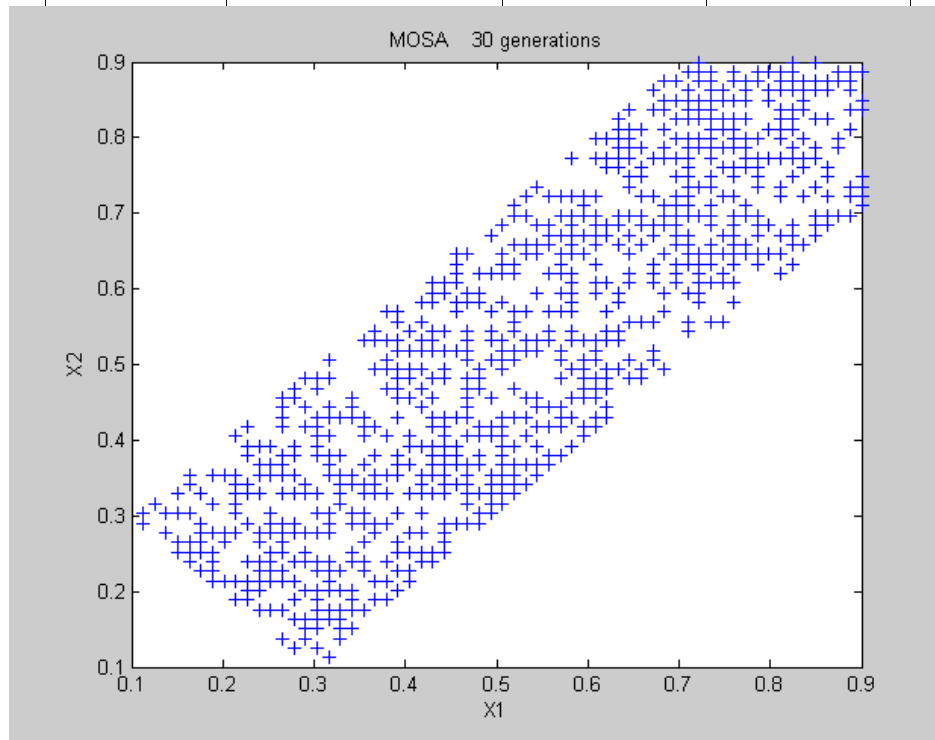
Mean	Standard deviation	PW-metric	CM-metric
0.014258	0.004751	0.333216	1.6566



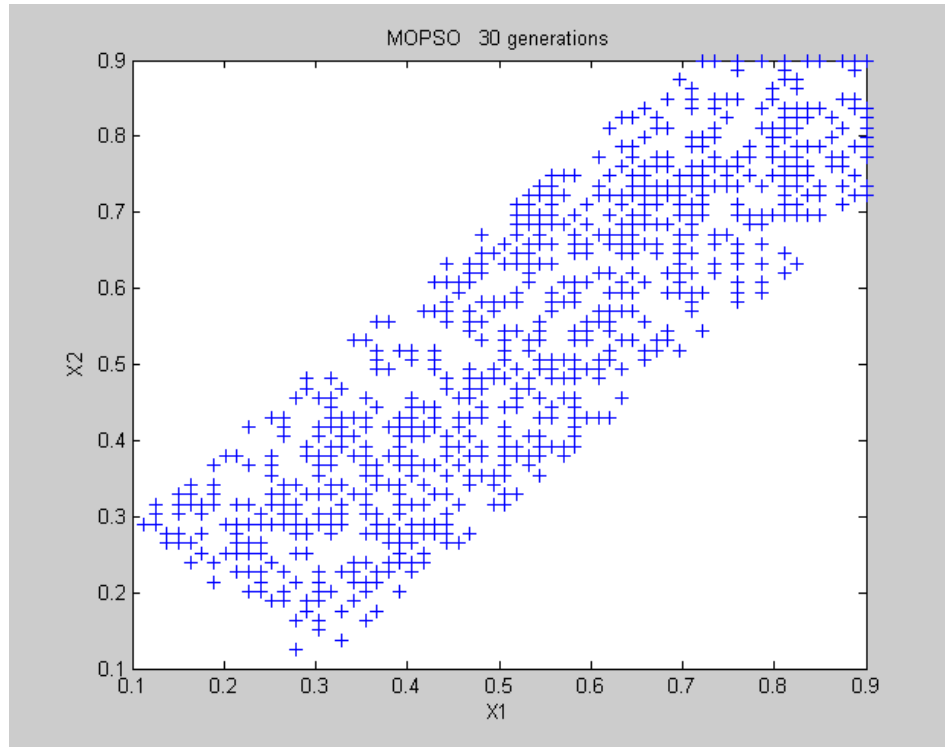
Mean	Standard deviation	PW-metric	CM-metric
0.006532	0.004273	0.654164	1.6542



Mean	Standard deviation	PW-metric	CM-metric
0.014126	0.003968	0.2809	1.6505



Mean	Standard deviation	PW-metric	CM-metric
0.013225	0.00209	0.158526	1.6498



Mean	Standard deviation	PW-metric	CM-metric
0.013471	0.002566	0.190483	1.6505

APPENDIX B

Abaqus Python Script

The main parts of the Python script are the following:

13. Initialisation (import required modules).
14. Define the Geometrical Parameters of the Joints
15. Define the x and y Coordinates of Main Points
16. Create the Part
17. Define the Materials
18. Create Section and Section Assignment; Define the Materials
19. Create the Assembly
20. Create Steps
21. Apply Load
22. Apply Boundary Conditions
23. Mesh
24. Create and Run the Job

1- Initialisation (Import Required Module)

The code regarding this step is listed below:

```
##### Import Abaqus Objects #####  
from abaqus import *  
from abaqusConstants import *  
import regionToolset  
session.viewports['Viewport: 1'].setValues(displayedObject=None)
```

```
from abaqus import *
```

The aim of this statement is to import the Abaqus module and create a reference to the object defined by that module.

```
from abaqusConstants import *
```

This statement is to make all the symbolic constants available to the script.

To black out the viewport, note that the viewport is the window using by Abaqus / CAE to displays the part by using this statement:

```
session.viewports['Viewport: 1'].setValues(displayedObject=None)
```

2- Define the Geometrical Parameters of the Joints

In this section of the script the geometrical parameters are defined first, which include the weld toe radius, weld toe angle, weld reinforcement and welded joint length, then the elastic properties of material and load condition. Additionally, the parameters which are defined in equations (6.1 to 6.4) are also included. The code regarding this part is listed below:

```
## ##### Geometrical Parameters for welded joint #####
#####
D4=tRadu=1.8  ## Upper Weld toe radius
D14=tRadd=1.8      ## Lower Weld toe radius
D5=tAnglu=14      ## Upper Weld toe angle
D15=tAngld=14     ## Lower Weld toe angle
D10=wThick_t=6    ## weld thickness (tr)
F10=wThick_b=6    ## weld thickness (br)
D11=UOL_t=0.92    ## Upper Overlap
D12=LOL_t=0.92    ## Lower Overlap
F12=LOL_b=30      ## Joint Length
YoungsMod=207000  ## Young Modulus
PnRatio=0.3       ## Poison's ratio
Load1=200         ## Load 1st BC  Tensile load #
Load2=-200        ## Load 2nd BC  Bending load # the (-) sign indicates the moment on the left hand
side is in the anticlockwise direction
Bend=3           ## If it is "3" the User defined bending will be "Y/3"
E6=D6=tAnglu_r=pi/180*tAnglu ##Angle in rad
G6=F6=tAngld_r=pi/180*tAngld##Angle in rad
E7=D7=Ku=D4*tan(D6/2)
G7=F7=Kd=D14*tan(G6/2)
E8=D8=Pu=(0.5*F10)*tan(D6)
G8=F8=Pl=(0.5*F10)*tan(G6)
```

3- Define the x and y Coordinates of Main Points

After the definition of the geometrical parameters in the previous step and using equations (6.3 to 6.6) to define the parameters k and P, the relations which relate all these parameters to define the x and y coordinates of the points are listed below:

```
#####
##### CreatePoints():#####
pt=zeros((24,2),Float)
pt[1]=(0,0)
pt[2]=(0,D10/2.)
pt[3]=(0,-D10/2.)
pt[4]=((F12-(0.7*F10)),D10/2.)
pt[5]=((F12-(0.7*F10)-D7),D10/2.)
pt[6]=((F12-(0.7*F10)+(D7*cos(D6))),((D10/2)+(D7*sin(D6))))
pt[7]=((F12-(0.2*F10)),((D10/2.)+D8))
pt[8]=((F12),((D10/2.)+D11))
pt[9]=((F12),(-(D10/2.)-D12))
pt[10]=((F12-(0.2*F10)),(-(D10/2.)-G8))
pt[11]=((F12-(0.7*F10)+(G7*cos(G6))),(-(D10/2)-(G7*sin(G6))))
pt[12]=((F12-(0.7*F10)),(-D10/2.))
pt[13]=((F12-(0.7*F10)-G7),(-D10/2.))
return pt
```

4- Create the Part

The two statements:

```
import sketch
```

```
import part
```

It is difficult to create a sketch or a part without these important statements. Their aim is to provide access to the object related to sketches and parts. With the definition of the points in the previous section, it is possible to create a sketch by connecting these points by line and circular arc by using the **Line** and **Arc3Points** method. The next step after sketching is to create a two dimensional part apart. The code related to this process is listed below:

```
#####
##### Create the part #####
import sketch
import part
```

```
##### a) Creating sketch based on the input provided #####
s1 = mdb.models['Model-1'].ConstrainedSketch(name='WeldSketch', sheetSize=200.0)
g, v, d, c = s1.geometry, s1.vertices, s1.dimensions, s1.constraints
s1.setPrimaryObject(option=STANDALONE)
s1.Line(point1=pt[1], point2=pt[2])
s1.Line(point1=pt[2], point2=pt[5])
s1.Line(point1=pt[1], point2=pt[3])
s1.Line(point1=pt[3], point2=pt[13])
s1.Arc3Points(point1=pt[5], point2=pt[6], point3=pt[4])
s1.Arc3Points(point1=pt[6], point2=pt[8], point3=pt[7])
s1.Arc3Points(point1=pt[9], point2=(pt[11]), point3=pt[10])
s1.Arc3Points(point1=pt[11], point2=(pt[12]), point3=pt[13])
s1.Line(point1=pt[8], point2=pt[9])
#####
##### b) Creating part based on sketch #####
p = mdb.models['Model-1'].Part(name='WeldGeo_V1', dimensionality=TWO_D_PLANAR,
type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
p.BaseShell(sketch=s1)
s1.unsetPrimaryObject()
```

5- Define the Materials

```
import material
```

This statement provides access to models and objects related to materials. The definition of materials properties, such as Young's modulus and Poisson's ratio, are done by using this code

```
##### Create material #####
import material
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((YoungsMod, PnRatio), ))
```

6- Create Section and Section Assignment Define the Materials

```
import section
```

This statement is used to make section properties and methods accessible to the script, whilst the following statement


```
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-1', material='Material-1',
thickness=None)
```

is used to create a solid section. Additionally, the section assignment is done by using this statement:

```
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
offsetType=MIDDLE_SURFACE, offsetField="
```

```
##### Create section #####
import section
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-1', material='Material-1',
thickness=None)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
region = regionToolset.Region(faces=faces)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
##### section assignment #####
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
offsetType=MIDDLE_SURFACE, offsetField="
a1 = mdb.models['Model-1'].rootAssembly
a1.regenerate()
```

7- Create the Assembly

```
import assembly
```

This statement is used to give the script access to assembly methods and properties.

```
##### Create the assembly #####
import assembly
a = mdb.models['Model-1'].rootAssembly
a.DatumCsysByDefault(CARTESIAN)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
a.Instance(name='WeldGeo_V1-1', part=p, dependent=ON)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
adaptiveMeshConstraints=ON)
```

8- Create Steps

```
import step
```

To make the script able to access the step methods and properties it is preferable to use this statement. The code below is used to create two steps in addition to the initial step.

```
##### Create the step #####
import step
```

```
## Create the step 1
mdb.models['Model-1'].StaticStep(name='Step-1', previous='Initial')
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')
## Create the step 2
mdb.models['Model-1'].StaticStep(name='Step-2', previous='Step-1')
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-2')
```

9- Apply Load

```
region = regionToolset.Region(side1Edges=side1Edges1)
```

This statement is used to create a region using the side1Edge1 object. The purpose of this edge is to specify the region of load application. Actually, there are two load cases. The first one is the tensile load, whilst the second is the moment load. However, it is useful to specify the analytical field before applying this load. This process is done by using this statement:

```
mdb.models['Model-1'].ExpressionField(name='AnalyticalField-1', localCsys=None,
description="", expression=' Y /3')
```

The code for applying the loads is listed below:

```
##### Apply load #####
## First we need to select the edge
region = regionToolset.Region(side1Edges=side1Edges1)
## ##### Apply tensile load in step 1 #####
mdb.models['Model-1'].Pressure(name='Load-1', createStepName='Step-1',
region=region, distributionType=UNIFORM, field="", magnitude=Load1,
amplitude=UNSET)
## define analytical field for the moment load
mdb.models['Model-1'].ExpressionField(name='AnalyticalField-1', localCsys=None,
description="", expression=' Y /3')
## select the edge
```

```
region = regionToolset.Region(side1Edges=side1Edges1)
##### Apply the load 2 (moment ) in step 2 #####
mdb.models['Model-1'].Pressure(name='Load-2', createStepName='Step-2',
region=region, distributionType=FIELD, field='AnalyticalField-1',
magnitude=Load2, amplitude=UNSET)
```

10- Apply Boundary Conditions

The first step in applying boundary condition is to specify the edge where this condition is applied by using this statement:

```
region = regionToolset.Region(edges=edges1)
```

The next step is to apply the symmetry boundary condition:

```
mdb.models['Model-1'].XsymmBC(name='BC-1', createStepName='Initial',  
    region=region)
```

```
##### Apply symmetric boundary condition #####  
region = regionToolset.Region(edges=edges1)  
mdb.models['Model-1'].XsymmBC(name='BC-1', createStepName='Initial',  
    region=region)  
region = regionToolset.Region(edges=edges1)  
mdb.models['Model-1'].XsymmBC(name='BC-2', createStepName='Step-1',  
    region=region)  
region = regionToolset.Region(edges=edges1)  
mdb.models['Model-1'].XsymmBC(name='BC-3', createStepName='Step-2',  
    region=region)
```

11-Mesh

```
import mesh
```

This statement makes the method and properties of the mesh module accessible to script.

```
##### Create the Mesh#####  
import mesh  
  
pickedFaces = f.getSequenceFromMask(mask=('[#1 ]', ), )  
v, e, d = p.vertices, p.edges, p.datums  
p.PartitionFaceByShortestPath(faces=pickedFaces, point1=p.InterestingPoint(edge=e[7], rule=MIDDLE),  
    point2=p.InterestingPoint(edge=e[1], rule=MIDDLE))  
  
pickedEdges2 = e.getSequenceFromMask(mask=('[#82 ]', ), )  
p.seedEdgeByBias(end2Edges=pickedEdges2, ratio=40.0, number=60, constraint=FINER)  
p = mdb.models['Model-1'].parts['WeldGeo_V1']  
e = p.edges
```

```

pickedEdges = e.getSequenceFromMask(mask=['#44'], ), )
p.seedEdgeByNumber(edges=pickedEdges, number=11, constraint=FINER)

pickedEdges2 = e.getSequenceFromMask(mask=['#28'], ), )
p.seedEdgeByBias(end2Edges=pickedEdges2, ratio=40.0, number=25, constraint=FINER)

pickedEdges1 = e.getSequenceFromMask(mask=['#20'], ), )
p.seedEdgeByBias(end1Edges=pickedEdges1, ratio=40.0, number=25, constraint=FINER)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
p.seedPart(size=wThick_t/8.0, deviationFactor=0.1)

pickedEdges = e.getSequenceFromMask(mask=['#10'], ), )
p.PartitionEdgeByParam(edges=pickedEdges, parameter=0.5)
p.generateMesh()

```

12- Create and Run the Job

import job

This statement is to allow the script to access the job method. The code below is used to create a job named ‘Weld joint’ by using the **Job()** method, then submit the job by utilising the **submit()** method. To make sure the control lies with the script until the analysis is finished is possible by using the statement

```
mdb.job['Welded joint'].waitForCompilation()
```

The code which is used to perform the analysis is listed below:

```

##### Create and run the job #####
import job
## create the job
mdb.Job(name='Welded joint', model='Model-1', type=ANALYSIS, explicitPrecision=SINGLE,
        nodalOutputPrecision=SINGLE, description="",
        parallelizationMethodExplicit=DOMAIN, multiprocessingMode=DEFAULT,
        numDomains=1, userSubroutine="", numCpus=1, memory=50,
        describingmemoryUnits=PERCENTAGE, scratch="", echoPrint=OFF, modelPrint=OFF,
        contactPrint=OFF, historyPrint=OFF)
## Run the job
mdb.jobs['Welded joint'].submit(consistencyChecking=OFF)

```

```
mdb.job['Welded joint'].waitForCompilation()  
## End of run job
```

Python script of welded joint

“Weld_Script_v2_3pointArc_New.py”

```

##### Import Abaqus Objects #####
from abaqus import *
from abaqusConstants import *
import sketch
from Numeric import*
import section
import regionToolset
import displayGroupMdbToolset as dgm
##### Import modules #####
import part
import material
import assembly
import step
import interaction
import load
import mesh
import job
import sketch
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior
##### open CAE model#####
openMdb(pathName='C:/Temp/Arc_geo.cae')
session.viewports['Viewport: 1'].setValues(displayedObject=None)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
#####
##### Geometrical Parameters for weld optimization#####
#####
D4=tRadu=0.8    ## Upper Weld toe radius
D14=tRadd=0.8   ## Lower Weld toe radius
D5=tAnglu=19    ## Upper Weld toe angle
D15=tAngld=19   ## Lower Weld toe angle
D10=wThick_t=6  ## weld thickness (tr)
F10=wThick_b=6  ## weld thickness (br)
D11=UOL_t=1.1   ## Upper Overlap
D12=LOL_t=1.1   ## Lower Overlap
F12=LOL_b=30    ## Lower Overlap
YoungsMod=207000 ## Youngs Modulus
PnRatio=0.3     ## Poison's ratio
Load1=-150      ## Load 1st BC   Tensile load #
Load2=-150      ## Load 2nd BC   Bending load #
Bend=3          ## If it is "3" the User defined bending will be "Y/3"
##derived
E6=D6=tAnglu_r=pi/180*tAnglu    ##Angle in rad
G6=F6=tAngld_r=pi/180*tAngld    ##Angle in rad
E7=D7=Ku=D4*tan(D6/2)
G7=F7=Kd=D14*tan(G6/2)
E8=D8=Pu=(0.5*F10)*tan(D6)
G8=F8=Pl=(0.5*F10)*tan(G6)
#####

```

```

#def CreatePoints():
pt=zeros((24,2),Float)
pt[0]=(0,0)
pt[1]=(0,D10/2.)
pt[2]=(0,-D10/2.)
pt[3]=((F12-(0.7*F10)),D10/2.)
pt[4]=((F12-(0.7*F10)-D7),D10/2.)
pt[5]=((F12-(0.7*F10)+(D7*cos(D6))),((D10/2)+(D7*sin(D6))))
pt[6]=((F12-(0.2*F10)),((D10/2.)+D8))
pt[7]=((F12),((D10/2.)+D11))
pt[8]=((F12+(0.2*F10)),((D10/2.)+E8))
pt[9]=((F12+(0.7*F10)-(E7*cos(E6))),((D10/2)+(E7*sin(E6))))
pt[10]=((F12+(0.7*F10)),(D10/2.))
pt[11]=((F12+(0.7*F10)+E7),(D10/2.))
pt[12]=((F12+F12),(D10/2.))
pt[13]=((F12+F12),(0))
pt[14]=((F12+F12),(-D10/2.))
pt[15]=((F12+(0.7*F10)+F7),(-D10/2.))
pt[16]=((F12+(0.7*F10)),(-D10/2.))
pt[17]=((F12+(0.7*F10)-F7*cos(F6)),(-(D10/2)-(F7*sin(F6))))
pt[18]=((F12+(0.2*F10)),(-(D10/2.)-F8))
pt[19]=((F12),(-(D10/2.)-D12))
pt[20]=((F12-(0.2*F10)),(-(D10/2.)-G8))
pt[21]=((F12-(0.7*F10)+(G7*cos(G6))),(-(D10/2)-(G7*sin(G6))))
pt[22]=((F12-(0.7*F10)),(-D10/2.))
pt[23]=((F12-(0.7*F10)-G7),(-D10/2.))
# return pt
#def WeldGeometryCreation(pt):#####
##### Creating sketch based on the input provided #####
s1 = mdb.models['Model-1'].ConstrainedSketch(name='WeldSketch', sheetSize=
200.0)
g, v, d, c = s1.geometry, s1.vertices, s1.dimensions, s1.constraints
s1.setPrimaryObject(option=STANDALONE)
#s1.Line(point1=pt[0], point2=pt[1])
s1.Line(point1=pt[1], point2=pt[4])
s1.Line(point1=pt[1], point2=pt[2])
s1.Line(point1=pt[2], point2=pt[23])
s1.Arc3Points(point1=pt[4], point2=pt[5], point3=pt[3])
s1.Arc3Points(point1=pt[5], point2=pt[7], point3=pt[6])
s1.Arc3Points(point1=pt[19], point2=(pt[21]), point3=pt[20])
s1.Arc3Points(point1=pt[21], point2=(pt[23]), point3=pt[22])
s1.Line(point1=pt[7], point2=pt[19])
#####
##### Creating part based on sketch #####
#####
p = mdb.models['Model-1'].Part(name='WeldGeo_V1', dimensionality=TWO_D_PLANAR
, type=DEFORMABLE_BODY)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
p.BaseShell(sketch=s1)
s1.unsetPrimaryObject()
#####
#def Mesh():
p = mdb.models['Model-1'].parts['WeldGeo_V1']

```

```

f = p.faces
pickedFaces = f.getSequenceFromMask(mask=('[#1 ]', ), )
v, e, d = p.vertices, p.edges, p.datums
p.PartitionFaceByShortestPath(faces=pickedFaces, point1=p.InterestingPoint(
edge=e[7], rule=MIDDLE), point2=p.InterestingPoint(edge=e[1], rule=MIDDLE))
p = mdb.models['Model-1'].parts['WeldGeo_V1']
e = p.edges
pickedEdges2 = e.getSequenceFromMask(mask=('[#82 ]', ), )
p.seedEdgeByBias(end2Edges=pickedEdges2, ratio=40.0, number=60, constraint=
FINER)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
e = p.edges
pickedEdges = e.getSequenceFromMask(mask=('[#44 ]', ), )
p.seedEdgeByNumber(edges=pickedEdges, number=11, constraint=FINER)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
e = p.edges
pickedEdges2 = e.getSequenceFromMask(mask=('[#28 ]', ), )
p.seedEdgeByBias(end2Edges=pickedEdges2, ratio=40.0, number=25, constraint=
FINER)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
e = p.edges
pickedEdges1 = e.getSequenceFromMask(mask=('[#20 ]', ), )
p.seedEdgeByBias(end1Edges=pickedEdges1, ratio=40.0, number=25, constraint=
FINER)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
p.seedPart(size=wThick_t/8.0, deviationFactor=0.1)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
e = p.edges
pickedEdges = e.getSequenceFromMask(mask=('[#10 ]', ), )
p.PartitionEdgeByParam(edges=pickedEdges, parameter=0.5)
p.generateMesh()
#####
#def Material():#####
##### Create material #####
mdb.models['Model-1'].Material(name='Material-1')
mdb.models['Model-1'].materials['Material-1'].Elastic(table=((YoungsMod,
PnRatio), ))
mdb.models['Model-1'].HomogeneousSolidSection(name='Section-1', material=
'Material-1', thickness=None)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
f = p.faces
faces = f.getSequenceFromMask(mask=('[#3 ]', ), )
region = regionToolset.Region(faces=faces)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
p.SectionAssignment(region=region, sectionName='Section-1', offset=0.0,
offsetType=MIDDLE_SURFACE, offsetField='')
a1 = mdb.models['Model-1'].rootAssembly
a1.regenerate()

#####
#####
a = mdb.models['Model-1'].rootAssembly          ###
Assembly- instance#####

```



```

session.viewports['Viewport: 1'].setValues(displayedObject=a)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    adaptiveMeshConstraints=OFF)
a = mdb.models['Model-1'].rootAssembly
a.DatumCsysByDefault (CARTESIAN)
p = mdb.models['Model-1'].parts['WeldGeo_V1']
a.Instance(name='WeldGeo_V1-1', part=p, dependent=ON)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(
    adaptiveMeshConstraints=ON)
mdb.models['Model-1'].StaticStep(name='Step-1', previous='Initial')
#####Create step 1#####
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')
mdb.models['Model-1'].StaticStep(name='Step-2', previous='Step-1')
##### Create step 2#####
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-2')
mdb.models['Model-1'].FieldOutputRequest(name='F-Output-2',
##### Create Field-output 2 with step 2
    createStepName='Step-2', variables=('S', 'MISESMAX', 'TSHR', 'CTSHR',
    'ALPHA', 'VS', 'PS'))
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Initial')
session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=ON, bcs=ON,
    predefinedFields=ON, connectors=ON, adaptiveMeshConstraints=OFF)
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['WeldGeo_V1-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#30 ]', ), )
region = regionToolset.Region(edges=edges1)
mdb.models['Model-1'].XsymmBC(name='BC-1', createStepName='Initial',
##### BC1 with initial step #####
    region=region)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-1')
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['WeldGeo_V1-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#30 ]', ), )
region = regionToolset.Region(edges=edges1)
mdb.models['Model-1'].XsymmBC(name='BC-2', createStepName='Step-1',
##### BC2 with step 1 #####
    region=region)
a = mdb.models['Model-1'].rootAssembly
s1 = a.instances['WeldGeo_V1-1'].edges
side1Edges1 = s1.getSequenceFromMask(mask=('[#400 ]', ), )
region = regionToolset.Region(side1Edges=side1Edges1)
mdb.models['Model-1'].Pressure(name='Load-1', createStepName='Step-1',
##### Load 1 with step 1 #####
    region=region, distributionType=UNIFORM, field='', magnitude=Load1, ###
    Load 1
    amplitude=UNSET)
session.viewports['Viewport: 1'].assemblyDisplay.setValues(step='Step-2')
a = mdb.models['Model-1'].rootAssembly
e1 = a.instances['WeldGeo_V1-1'].edges
edges1 = e1.getSequenceFromMask(mask=('[#30 ]', ), )
region = regionToolset.Region(edges=edges1)
mdb.models['Model-1'].XsymmBC(name='BC-3', createStepName='Step-2',
##### BC 3 with step 2 #####

```

```

        region=region)
mdb.models['Model-1'].ExpressionField(name='AnalyticalField-1', localCsys=
None, ## Load 2 with step 3 #####
    description='', expression=' Y /3')
    Y/3 #####
a = mdb.models['Model-1'].rootAssembly
s1 = a.instances['WeldGeo_V1-1'].edges
sideEdges1 = s1.getSequenceFromMask(mask=('[#400 ]', ), )
region = regionToolset.Region(sideEdges=sideEdges1)
mdb.models['Model-1'].Pressure(name='Load-2', createStepName='Step-2',
    region=region, distributionType=FIELD, field='AnalyticalField-1',
    magnitude=Load2, amplitude=UNSET)

##### PRESSURE VALUE

#####
mdb.models['Model-1'].loads['Load-1'].deactivate('Step-2')
mdb.models['Model-1'].boundaryConditions['BC-1'].deactivate('Step-1')
mdb.models['Model-1'].boundaryConditions['BC-2'].deactivate('Step-2')
session.viewports['Viewport: 1'].assemblyDisplay.setValues(loads=OFF, bcs=OFF
,
    predefinedFields=OFF, connectors=OFF)
mdb.Job(name='Job-1', model='Model-1', type=ANALYSIS, explicitPrecision=
SINGLE,
    nodalOutputPrecision=SINGLE, description='',
    parallelizationMethodExplicit=DOMAIN, multiprocessingMode=DEFAULT,
    numDomains=1, userSubroutine='', numCpus=1, memory=50,
    memoryUnits=PERCENTAGE, scratch='', echoPrint=OFF, modelPrint=OFF,
    contactPrint=OFF, historyPrint=OFF)
mdb.jobs['Job-1'].submit(consistencyChecking=OFF)

```

APPENDIX C

Example calculation for fatigue test experiment

By using stress ratio ($R=0.1$), $R = \frac{\sigma_{min}}{\sigma_{max}}$

$$\sigma_{min} = 0.1 \sigma_{max} \dots\dots\dots (1)$$

$$\sigma_{mean} = \frac{\sigma_{max} + \sigma_{min}}{2} = \frac{1.1 \sigma_{max}}{2} = 0.55 \sigma_{max} \dots\dots\dots (2)$$

$$\Delta\sigma = \sigma_{range} = \sigma_{max} - \sigma_{min} = \sigma_{max} - 0.1\sigma_{max} = 0.9 \sigma_{max} \dots\dots\dots (3)$$

From equations 2 and 3

$$\begin{aligned} \sigma_{mean} &= 0.55 \sigma_{max} & \sigma_{max} &= \frac{\Delta\sigma}{0.9} \\ \sigma_{mean} &= 0.55 \frac{\Delta\sigma}{0.9} = 0.611 \Delta\sigma \dots\dots\dots (4) \end{aligned}$$

For $\Delta\sigma = 310 \text{ MPa}$ and the dimension of the specimen (19.16 * 6.3) mm

$$\sigma_{mean} = 0.611 * 310 = 189.41 \text{ MPa}$$

$$\sigma_{amp} = \frac{\Delta\sigma}{2} = \frac{310}{2} = 155 \text{ MPa}$$

$$F_{mean} = \sigma_{mean} * A = 189.41 * 19.16 * 6.3 = 22.863 \text{ kN}$$

$$F_{amp} = \sigma_{amp} * A = 155 * 19.16 * 6.3 = 18.709 \text{ kN}$$

Stress range ($\Delta\sigma$) MPa	Mean stress (σ_{mean})MPa	Stress amplitude (σ_{amp})MPa	Dimension of specimen		Mean load (F_{mean})kN	Load amplitude (F_{amp})kN
			Width (mm)	Height (mm)		
200	122.1	100	19.16	6.2	14.516	11.879
225	137.47	112.5	19.2	6.3	16.629	13.608
250	152.75	125	19.1	6.3	18.380	15.041
275	168.02	137.5	19.3	6.4	20.754	16.984
310	189.41	155	19.16	6.3	22.863	18.710
325	198.58	162.5	19.3	6.5	24.911	20.386